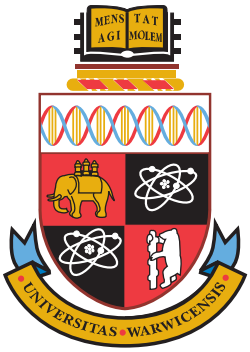


Bayesian inference for multi-level non-stationary Gaussian processes

by

Karla Monterrubio Gómez

Thesis
Submitted to the University of Warwick
for the degree of

Doctor of Philosophy in Statistics

University of Warwick
Department of Statistics

September 2019

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ALGORITHMS

ACKNOWLEDGMENTS

First, I thank Prof. Mark Girolami for giving me the opportunity of pursuing this degree and welcoming me to his research team. Also, I want to acknowledge my second supervisor, Dr Theo Damoulas, for his time, helpful feedback, support and guidance during these years.

I was fortunate to meet Dr Sara Wade, to whom I want to express my deepest and everlasting gratitude, for being a great mentor and most importantly, a caring friend. I have learned immensely from her approach to do research and her critical thinking. After every meeting and discussion with her I find myself more motivated. I definitely would have not being able to write this thesis without her invaluable support, guidance, patience, knowledge and extraordinary commitment. Thank you Sara for all the time you have spent working with me, you have made this journey way better.

Also, very special thanks to Dr Lassi Roninen for the opportunities, feedback, interesting discussions and for always suggesting new ideas and projects. In particular, I am grateful for all the SOS skype sessions, where he always lent a helping hand. It has been a truly enjoyable experience to collaborate with him.

In addition, I want to thanks my examiners, Prof Gareth Roberts and Prof Magnus Rattray, for their feedback, suggestions and questions that helped me to improve this work and enriched my research perspective. I truly had an excellent time during my examination.

I also want to sincerely thank Dr Carlos Cuevas for encouraging me to continue studying, and whose passion for statistics start shaping my career path.

Importantly, I extend my gratitude to all the amazing people I met during these years and who gave me a family far from home. Thanks to Ale and Elia for always being there for me when I needed the most; to Felipe and Moni for making this stage funnier; to my craziest and sweetest friend: Beni; and to Adam for his inmense patience, all the maths and computing lessons, and for staying by my side during the hard times. Thanks also to those friends that have stayed with me despite time and distance, cheering me up and making me feel close to home: Aurea, Diego, Sama, Angie and Karla.

Por supuesto, gracias a mi familia que desde lejos me han apoyado en todas las formas posibles y han sido y serán siempre mi más grande fuerza. Lo logramos!

DECLARATION

I hereby declare that I have written this PhD thesis completely by myself, under the supervision of Dr Theo Damoulas, Prof. Mark Girolami and Dr Sara Wade. The thesis is submitted to the University of Warwick for the degree of Doctor of Philosophy in Statistics, and I confirm that the work here presented has not been submitted for a degree at any other university.

Chapter 4 of this thesis is based on the content of the article **"Posterior Inference for Sparse Hierarchical Non-stationary Models"** (Monterrubio-Gómez et al., 2019), which is a result of collaborative work with Dr Sara Wade, Dr Lassi Roininen, Dr Theo Damoulas and Prof. Mark Girolami. This article has been submitted to a peer-reviewed journal and is under revision. A preprint of this article can be found at `https://arxiv.org/abs/1804.01431`.

Finally, I declare that I have not used sources or means without declaration in the text.

ABSTRACT

The complexity of most real-world phenomena requires the use of flexible models that capture intricated features present in the data. Gaussian processes (GPs) have proven valuable tools for this purpose due to their non parametric and probabilistic nature. Nevertheless, the default approach when modelling with GPs is to assume stationarity. This assumption permits easier inference but can be restrictive when the correlation of the process is not constant across the input space.

This thesis investigates a class of non-stationary priors that enhance flexibility while retaining interpretability. These priors assemble GPs through input-varying parameters in the covariance. Such hierarchical constructions result in high-dimensional correlated posteriors, where Bayesian inference becomes challenging and notably expensive due to the characteristic computational constrains of GPs. Altogether, this thesis provides novel approaches for scalable Bayesian inference in 2-level GP regression models. First, we use a sparse representation of the inverse non-stationary covariance to develop and compare three different Markov chain Monte Carlo (MCMC) samplers for two hyperpriors. To maintain scalability when extending the approach to multi-dimensional problems, we propose a non-stationary additive Gaussian process (AGP) model. The efficiency and accuracy of the methodology are demonstrated in simulated experiments and a computer emulation problem. Second, we derive a hybrid variational-MCMC approach that combines low-dimensional variational distributions with MCMC to avoid further distributional and independence restrictions on the posterior of interest. The resulting approximate posterior includes an intractable likelihood that when approximated with a small-order Gauss-Hermite quadrature results in poor predictive performance. In this case, an extension to higher-dimensional settings requires specific assumptions of the non-stationary covariance. Lastly, we propose a pseudo-marginal algorithm that uses a block-Poisson estimator to circumvent numerical integration in the variationally sparse model. This strategy demonstrates an improvement in predictive performance, can be computationally more efficient, and is generally applicable to other GP-based models with intractable likelihoods.

CHAPTER 1

INTRODUCTION

1.1 Motivation

Many datasets and real-world applications present complexities that challenge standard statistical models. Under a parametric setting, models are parametrised by a finite set of parameters, making inference and interpretation straightforward. However, this parametric assumption can result in models with limited flexibility that can fail to recover essential structures in the data. Non parametric approaches overcome this issue by working over infinite-dimensional parameter spaces, thus, enhancing model flexibility.

The Bayesian approach to modelling arises through the interpretation of probabilities as ones subjective beliefs; consequently, one can construct a prior distribution over the parameter space to reflect uncertainty and encapsulate any prior knowledge. In Bayesian non-parametrics (BNP), this involves employing stochastic processes as prior distributions over the infinite-dimensional parameter space. However, defining a non-parametric prior is delicate, and in general, such a prior should have (i) large support, (ii) interpretable hyperparameters and (iii) tractable posterior inference. The Gaussian process (GP) prior is one of the most popular choices in BNP precisely because it satisfies these criteria. The focus of this thesis is on flexible Bayesian non-parametric models constructed by stacking GP priors.

In the literature, GPs are frequently utilised in constructing powerful models in a wide range of applications. GP priors have been used in geostatistics (Matheron, 1973) under the name of Kriging. They are also common in other applications; for instance, in atmospheric sciences (Berrocal et al., 2010), biology (Stathopoulos et al., 2014), genetics (Rattray et al., 2019), and inverse problems (Kaipio and Somersalo,

2006).

A large amount of research on GPs and their applications has focused on models where an assumption of stationarity for the process of interest is made. Heaton et al. (2018) provide a complete review and comparison of available methods under this assumption. Nevertheless, this assumption is rarely realistic in practice and as a consequence, several approaches to introduce non-stationarity have been proposed (e.g. Anderes and Stein, 2008; Gramacy and Lee, 2012; Kim et al., 2005; Montagna and Tokdar, 2016; Sampson et al., 2001). Although comparative evaluations show that removing the stationary assumption improves predictive accuracy (Gramacy and Lee, 2012; Neto et al., 2014; Fouedjio et al., 2016), fitting such non-stationary models has proven to be challenging. This, combined with the well-known computational constraints of GP models, arising from storing covariance matrices, solving linear systems and computing determinants, poses important questions on how to efficiently perform Bayesian inference in non-stationary problems.

Our interest is in studying extensions of standard (one-level or single-level) GP priors to more complicated but interpretable hierarchical structures that stack GP priors through spatially varying parameters in the covariance function. We refer to these priors as multi-level GP priors. More precisely, this thesis aims to develop efficient Bayesian inference algorithms for such models, thus creating novel inference tools with a more realistic underlying assumption that can serve to model various real-world problems more accurately.

1.2 Outline of the thesis

We begin in Chapter 2 with a comprehensive review of the theoretical background behind Gaussian process (GP) models. The chapter highlights the relevance of the covariance function in defining the properties of the process and lists some common covariance function choices. Also, we include a brief overview of Markov chain Monte Carlo (MCMC) methods with a focus on GP inference. The chapter finishes by introducing the models of interest in this thesis; namely, multi-level non-stationary GPs and discusses their connection to similar hierarchical constructions in the literature.

Chapter 3 investigates the main challenges that arise when doing fully Bayesian inference in 2-level Gaussian process regression (GPR) settings. These challenges include (i) devising effective MCMC samplers, (ii) parameter identifiability, and (iii) efficient scalability in both the number of data points and dimensions. Firstly, we propose to utilise elliptical slice sampling (Ell-SS) to sample the spatially varying

parameter. Secondly, we propose the usage of empirical priors to improve parameter recovery and identifiability by constraining the parameters according to the observed data. Thirdly, the chapter introduces a parametrisation of the spatially varying parameter based on LDL factorisation that provides information about the range and direction of dependence of the process, but can only be applied effectively to moderate-size datasets with a small number of dimensions. Finally, the work presented in Chapter 4 and Chapter 5 address the computational burden of the model from different perspectives.

On the one hand, Chapter 4 considers a Gaussian Markov random field (GMRF) formulation of the model, where the sparse and banded structure of the finite-dimensional approximation of the inverse non-stationary covariance alleviates the computational constraints of doing exact inference. The work examines and compares (in mixing performance and computational cost) three different sampling methodologies under different prior assumptions of the correlation process. Furthermore, it introduces a novel extension of the model for multi-dimensional problems based on additive GPs. Importantly, the proposed additive approach is scalable and interpretable but also retains flexibility. The capabilities of the method are demonstrated in both simulated experiments and a computer emulation problem.

On the other hand, Chapter 5 adopts an inducing variable approach and variational inference to derive optimal, free-form, low-dimensional, approximate posterior distributions. Importantly, this variational approach is combined with MCMC to effectively explore the posterior, without imposing the additional further constrains, e.g. independence, that are typically required in full variational Bayes schemes. The derivations are provided for three different formulations of the non-stationary covariance function in multi-dimensional input spaces, that correspond to different assumptions of the correlation structure of the process. In all three cases, the posterior of interest contains an expected log-likelihood term that is intractable. Our experiments and analyses suggest that approximating the required expectations with Gauss-Hermite quadrature can be inaccurate, especially when employing a small number of nodes.

Following this and making use of recent developments for scalable MCMC algorithms, Chapter 6, proposes to avoid numerical integration and instead utilise a pseudo-marginal scheme. The suggested approach employs data subsampling and an estimator constructed as a product of Poisson estimators. Initial results show that the methodology offers considerable computational gains with improved predictive performance. Importantly, the inference method can be applied more generally, and Chapter 6 concludes by enumerating some common GP models than can benefit

from this approach.

Finally, we conclude this thesis in Chapter 7 with a discussion about the main findings of this work and some possible extensions.

1.3 Research outcomes

(i) Chapter 4 is based on the content of the article **"Posterior Inference for Sparse Hierarchical Non-stationary Models"** (Monterrubio-Gómez et al., 2019), which is a result of collaborative work with Dr Sara Wade, Dr Lassi Roininen, Dr Theo Damoulas and Prof Mark Girolami. This article has been submitted to a peer-reviewed journal and is under revision. A preprint of this article can be found at `https://arxiv.org/abs/1804.01431`.

(ii) The work introduced in Chapter 5 and Chapter 6 will result in an article titled **"On MCMC for variationally sparse Gaussian processes: A pseudo-marginal approach"** (under preparation). The article will be submitted to the Journal of Machine Learning Research. This work is in collaboration with Dr Sara Wade.

(iii) The code employed for (i) and (ii) will be made publicly available with the manuscripts, and also via a GitHub repository. This will ensure reproducibility and provide full transparency for applications and developing novel algorithms. In addition, we aim to develop an R package for fast inference in 2-level GP models.

CHAPTER 2

REVIEW

Bayesian nonparametrics employs stochastic processes as prior distributions. In contrast to parametric models, which assume a finite number of parameters, non-parametric models work over infinite-dimensional parameter space. The nature of these models makes them extremely flexible and therefore widely applicable. The focus of this thesis is on one of this classes of models; namely, Gaussian processes.

Gaussian processes are related to many other statistical models and machine learning algorithms, such as generalised linear regression (Rasmussen and Williams, 2006, Chapter 2), neural networks (Neal, 1995), spline models and support vector machines (Seeger, 2000; Sollich, 2002). Moreover, this family of models can be employed to perform several tasks, including classification, interpolation, dimensionality reduction (Lawrence, 2004), optimisation (Buche et al., 2005), integral approximations (O'Hagan, 1991), among others.

This chapter provides an introduction to Gaussian processes (GPs), highlighting the importance of the covariance function in such models. It also provides a review on Markov chain Monte Carlo inference methods with a focus on GP models. Finally, the last section of this chapter introduces non-stationary multi-level Gaussian processes.

2.1 Fundamentals of Gaussian processes

Definition 2.1. A GP is a stochastic process $\{z(\boldsymbol{x}) : \boldsymbol{x} \in \mathcal{X} \subset \mathbb{R}^D\}$ whose finite-dimensional distributions are multivariate Gaussian distributions, i.e. for all $N \in \mathbb{N}$ and any $\boldsymbol{x}_1, ..., \boldsymbol{x}_N$, $\boldsymbol{x}_n \in \mathcal{X}$, the joint distribution of $z(\boldsymbol{x}_1), ..., z(\boldsymbol{x}_N)$ is an N-dimensional Gaussian distribution with consistent parameters.

Therefore, a GP is a collection of random variables, that intuitively can be seen as a generalisation of the multivariate Gaussian distribution to infinite index sets (Seeger, 2004). As with a multivariate Gaussian, the properties of a GP are completely characterised by its mean and covariance functions,

$$\begin{aligned} \mathbb{E}[z(\boldsymbol{x}_i)] &= \mu(\boldsymbol{x}_i), \\ \mathrm{Cov}[z(\boldsymbol{x}_i), z(\boldsymbol{x}_j)] &= C(\boldsymbol{x}_i, \boldsymbol{x}_j), \end{aligned} \tag{2.1}$$

for any $\boldsymbol{x}_i$, $\boldsymbol{x}_j \in \mathcal{X} \subset \mathbb{R}^D$. We denote such distribution as

$$z(\cdot) \sim \mathrm{GP}\left(\mu(\cdot), C(\cdot, \cdot)\right).$$

When modelling, GPs are employed as prior distributions over the space of functions, and the mean and covariance are specified to reflect prior knowledge and assumptions of the unknown function.

When the mean function is set at zero, such that $\mu(\boldsymbol{x}) = 0$ for all $x \in \mathcal{X}$, the process is called a zero-centred or zero-mean Gaussian process. Zero-mean GPs are a common choice in the literature because this simplifies calculations and makes the properties of the process completely determined by the covariance function (Rasmussen and Williams, 2006). Note that if this assumption appears unrealistic for the observed data, one can simply subtract the mean of the response to center it at zero. Moreover, if we are interested in local properties of the process, employing a zero-mean GP is not a severe limitation because the posterior mean will not be necessarily zero. Because the focus of this thesis is on second-order non-stationary models for interpolation, we work only with constant mean GPs. Nevertheless, we emphasise that a non-constant mean function can be required in certain applications, for instance, extrapolation. An approach on how to specify non-constant mean functions based on a linear model is described in Rasmussen and Williams (2006, Chapter 2), and more generally, low-order polynomials are sometimes used (Stein, 2012).

The covariance function encodes essential properties of the process, such as its variation and smoothness. This function should define a valid covariance matrix (symmetric and positive semi-definite) for any finite set $X = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N)^T \subseteq \mathcal{X}$, which is denoted by C_X and has entries $C(\boldsymbol{x}_i, \boldsymbol{x}_j)$. To simplify the notation, we drop the subscript X, denoting the covariance matrix by C when the context is clear. Furthermore, for constant mean processes, the properties of the covariance function allow us to distinguish between two types of GPs, stationary and non-

stationary Gaussian processes. In Section 2.1.2, we define these types of GPs, and we depict realisations employing some covariance functions available in the literature. Throughout this thesis, we also refer to the covariance function as the kernel function, or only as the kernel.

2.1.1 Gaussian process modelling

This section explores how to construct models and do inference and predictions employing Gaussian processes. Let $\boldsymbol{y} = (y_1, \ldots, y_N)^T$ with $y_n \in \mathbb{R}$ denote the target or response variable, $X \in \mathbb{R}^{N \times D}$ be the matrix of input covariates or input locations, where the entry x_{nd} is the n^{th} observation corresponding to the d^{th} dimension ($n = 1, \ldots, N$, and $d = 1, \ldots D$), and $\boldsymbol{x}_n$ corresponds to the n^{th} row of X. A fully Bayesian version of a GP model can be written in a hierarchical form,

$$\begin{aligned} y_n &\sim p(y_n \mid z(\boldsymbol{x}_n), \boldsymbol{\rho}), \quad n = 1, ..., N, \\ z(\cdot) &\sim \text{GP}\left(\mu, C_{\boldsymbol{\phi}}\big(\cdot, \cdot\big)\right), \\ (\boldsymbol{\phi}, \boldsymbol{\rho}) &\sim \pi(\boldsymbol{\phi})\pi(\boldsymbol{\rho}), \end{aligned} \tag{2.2}$$

where $p(y_n \mid z(\boldsymbol{x}_n), \boldsymbol{\rho})$ is the likelihood, which is assumed to factorise across data points and depend on a latent function $z : \mathbb{R}^D \to \mathbb{R}$ that maps the input locations to the real line. Furthermore, $\boldsymbol{\rho}$ denotes the observation model parameters, and $\boldsymbol{\phi}$ denotes the parameters of the covariance function $C_{\boldsymbol{\phi}}(\cdot, \cdot)$ (see Section 2.1.2 for more details). Note that the response variable y_n depends on $\boldsymbol{x}_n$ only through the latent function z, and $z(\boldsymbol{x}_n)$ is the value of the latent function at $\boldsymbol{x}_n$. The key component of the hierarchical model is the GP prior over the function space in the second line of Eq. (2.2), where μ represents the a priori constant mean of the process.

Following a fully Bayesian approach, we are interested in the posterior distribution of all the unknowns in the model, the latent function, parameters, and hyperparameters,

$$\pi(\boldsymbol{z}, \boldsymbol{\rho}, \boldsymbol{\phi} \mid X, \boldsymbol{y}) = \pi(\boldsymbol{z} \mid X, \boldsymbol{y}, \boldsymbol{\rho}, \boldsymbol{\phi})\pi(\boldsymbol{\rho}, \boldsymbol{\phi} \mid X, \boldsymbol{y}), \tag{2.3}$$

where $\boldsymbol{z} = (z(\boldsymbol{x}_1), ..., z(\boldsymbol{x}_N))^T$ is a vector collecting function values at the observed locations. Firstly, we aim to calculate the conditional posterior distribution over the latent function $\pi(\boldsymbol{z} \mid X, \boldsymbol{y}, \boldsymbol{\rho}, \boldsymbol{\phi})$. Secondly, we target the marginal posterior distribution of the observation model parameters and hyperparameters in the model $\pi(\boldsymbol{\rho}, \boldsymbol{\phi} \mid X, \boldsymbol{y})$. By Bayes' rule, the conditional posterior distribution over the latent function is

$$\pi(\boldsymbol{z} \mid X, \boldsymbol{y}, \boldsymbol{\rho}, \boldsymbol{\phi}) = \frac{p(\boldsymbol{y} \mid \boldsymbol{z}, \boldsymbol{\rho})\pi(\boldsymbol{z} \mid X, \boldsymbol{\phi})}{\int p(\boldsymbol{y} \mid \boldsymbol{z}, \boldsymbol{\rho})\pi(\boldsymbol{z} \mid X, \boldsymbol{\phi})\, d\boldsymbol{z}}, \tag{2.4}$$

where $p(\boldsymbol{y} \mid \boldsymbol{z}, \boldsymbol{\rho}) = \prod_{n=1}^{N} p(y_n \mid z(\boldsymbol{x}_n), \boldsymbol{\rho})$ is the likelihood function, $\pi(\boldsymbol{z} \mid X, \boldsymbol{\phi})$ is the GP prior for z, and $\int p(\boldsymbol{y} \mid \boldsymbol{z}, \boldsymbol{\rho})\pi(\boldsymbol{z} \mid X, \boldsymbol{\phi})\, d\boldsymbol{z} = p(\boldsymbol{y} \mid X, \boldsymbol{\rho}, \boldsymbol{\phi})$, is called the *marginal likelihood*, *model evidence* or *type II likelihood*. Similarly, the marginal posterior distribution of the parameters and hyperparameters in the model is given by

$$\pi(\boldsymbol{\rho}, \boldsymbol{\phi} \mid X, \boldsymbol{y}) = \frac{p(\boldsymbol{y} \mid X, \boldsymbol{\rho}, \boldsymbol{\phi})\pi(\boldsymbol{\rho})\pi(\boldsymbol{\phi})}{\int p(\boldsymbol{y} \mid X, \boldsymbol{\rho}, \boldsymbol{\phi})\pi(\boldsymbol{\rho})\pi(\boldsymbol{\phi})\, d\boldsymbol{\rho}\, d\boldsymbol{\phi}}, \tag{2.5}$$

with the normalising constant $\int p(\boldsymbol{y} \mid X, \boldsymbol{\rho}, \boldsymbol{\phi})\pi(\boldsymbol{\rho})\pi(\boldsymbol{\phi})\, d\boldsymbol{\rho}\, d\boldsymbol{\phi} = p(\boldsymbol{y} \mid X)$. Notice that in this level, the marginal likelihood from Eq. (2.4) plays the role of the likelihood in Bayes' formula.

The simplest version of the GP model shown in Eq. (2.2) is obtained when the likelihood is assumed to be Gaussian (see Section 2.1.1.1). In this case, a closed-form is available for the marginal likelihood in the denominator of Eq. (2.4), as well as for the conditional posterior distribution over the latent function. When the likelihood is not Gaussian, the integral in the denominator at Eq. (2.4) is not analytically tractable, and one may restore to approximations of the conditional posterior of $\boldsymbol{z}$ (see Section 2.1.1.2).

When making inference, it is common practice to optimise the marginal likelihood (or an approximation of it) to obtain point estimates of the parameters $\boldsymbol{\rho}$ and hyperparameters $\boldsymbol{\phi}$. This is because solving the integral in Eq. (2.5) is in general difficult. This approach is known as *type II maximum likelihood* or *empirical Bayes* (Rasmussen and Williams, 2006).

However, a fully Bayesian treatment of the model considers prior distributions for $\boldsymbol{\rho}$ and $\boldsymbol{\phi}$, and we obtain the marginal posterior distribution over the latent function by integrating out parameters and hyperparameters, allowing us to account for uncertainty in the estimates. The marginal posterior is given by

$$\pi(\boldsymbol{z} \mid X, \boldsymbol{y}) = \int \pi(\boldsymbol{z} \mid X, \boldsymbol{y}, \boldsymbol{\rho}, \boldsymbol{\phi})\pi(\boldsymbol{\rho}, \boldsymbol{\phi} \mid X, \boldsymbol{y})\, d\boldsymbol{\rho}\, d\boldsymbol{\phi}. \tag{2.6}$$

Marginalisation over $\boldsymbol{\rho}$ and $\boldsymbol{\phi}$ can be done in several ways. The two most common approaches are *maximum a posteriori (MAP)* and *Monte Carlo methods*. Firstly, MAP approximates the integral by obtaining point estimates of the hyperparameters by maximising $\pi(\boldsymbol{\rho}, \boldsymbol{\phi} \mid X, \boldsymbol{y})$; therefore, $\pi(\boldsymbol{z} \mid X, \boldsymbol{y}) \approx \pi(\boldsymbol{z} \mid X, \boldsymbol{y}, \widehat{\boldsymbol{\rho}}, \widehat{\boldsymbol{\phi}})$ where $\widehat{\boldsymbol{\rho}}$ and $\widehat{\boldsymbol{\phi}}$ are the MAP estimates. Although this approach is straightforward to implement, it can underestimate the uncertainty (Vanhatalo et al., 2015). Secondly, Monte Carlo integration will allow us to numerically integrate Eq. (2.6) by drawing samples from $\pi(\boldsymbol{\rho}, \boldsymbol{\phi} \mid X, \boldsymbol{y})$. Likewise, when sampling from the marginal posterior of the

parameters and hyperparameters is not feasible, we can use Markov chain Monte Carlo (MCMC) methods to draw samples from the joint posterior of all the unknowns in the model; specifically, a Gibbs scheme (see Section 2.2) will alternate sampling from the full conditionals, $\pi(\boldsymbol{z} \mid X, \boldsymbol{y}, \boldsymbol{\rho}, \boldsymbol{\phi})$ and $\pi(\boldsymbol{\rho}, \boldsymbol{\phi} \mid X, \boldsymbol{y}, \boldsymbol{z})$. Alternatively, Rue et al. (2009) propose employing a central composite design (CCD) approach to select characteristic points from $\pi(\boldsymbol{z} \mid X, \boldsymbol{y}, \boldsymbol{\rho}, \boldsymbol{\phi})$ and employ them to approximate the integral as a weighted sum. The authors report that the method works well even for large parameter spaces, in contrast to a grid search methodology.

Now, consider making predictions of the latent function ("noise-free" predictions) at N^* new locations, $X^* \in \mathbb{R}^{N^* \times D}$, and denote $\boldsymbol{z}^* = (z(\boldsymbol{x}_1^*), \ldots, z(\boldsymbol{x}_{N^*}^*))^T$ the function value at the new inputs. Similar to inference, the marginal posterior predictive distribution for $\boldsymbol{z}^*$ is obtained by integrating out the parameters, the hyperparameters, and the latent function values; that is

$$\pi(\boldsymbol{z}^* \mid X, \boldsymbol{y}, X^*) = \int \pi(\boldsymbol{z}^* \mid X, X^*, \boldsymbol{z}, \boldsymbol{\rho}, \boldsymbol{\phi}) \pi(\boldsymbol{z}, \boldsymbol{\rho}, \boldsymbol{\phi} \mid X, \boldsymbol{y})\, d\boldsymbol{z}\, d\boldsymbol{\rho}\, d\boldsymbol{\phi}, \tag{2.7}$$

where $\pi(\boldsymbol{z}^* \mid X, X^*, \boldsymbol{z}, \boldsymbol{\rho}, \boldsymbol{\phi})$ is the conditional distribution obtained from the joint prior distribution of $\boldsymbol{z}$ and $\boldsymbol{z}^*$, which by definition[†] is normally distributed; namely,

$$\pi(\boldsymbol{z}^* \mid \boldsymbol{z}, X, X^*, \boldsymbol{\rho}, \boldsymbol{\phi}) = \mathrm{N}\left(\boldsymbol{\mu} + C_{\boldsymbol{\phi}}^* C_{\boldsymbol{\phi}}^{-1}(\boldsymbol{z} - \boldsymbol{\mu}), C_{\boldsymbol{\phi}}^{**} - C_{\boldsymbol{\phi}}^* C_{\boldsymbol{\phi}}^{-1} C_{\boldsymbol{\phi}}^{*T}\right), \tag{2.8}$$

with $\boldsymbol{\mu}$ an N-dimensional vector with μ in all its entries and $C_{\boldsymbol{\phi}}^*$ denoting a $(N^* \times N)$ cross-covariance matrix obtained by evaluating the covariance function $C_{\boldsymbol{\phi}}(\cdot, \cdot)$ for each pair $\boldsymbol{x}_i^*$ and $\boldsymbol{x}_j$. Similarly, $C_{\boldsymbol{\phi}}^{**}$ is an $(N^* \times N^*)$ covariance matrix, whereas $C_{\boldsymbol{\phi}}$ is of dimension $(N \times N)$. $C_{\boldsymbol{\phi}}^{*T}$ is the transpose of $C_{\boldsymbol{\phi}}^*$ and therefore an $(N \times N^*)$ matrix.

The integral in Eq. (2.7) is intractable because it requires the posterior distribution of all the unknowns in the model. Nevertheless, we can approximate it via Monte Carlo, through

$$\pi(\boldsymbol{z}^* \mid X, \boldsymbol{y}, X^*) \approx \frac{1}{T} \sum_{t=1}^{T} \pi(\boldsymbol{z}^* \mid \boldsymbol{z}^{(t)}, X, \boldsymbol{y}, X^*, \boldsymbol{\rho}^{(t)}, \boldsymbol{\phi}^{(t)}), \tag{2.9}$$

where $\boldsymbol{z}^{(t)}$, $\boldsymbol{\rho}^{(t)}$, and $\boldsymbol{\phi}^{(t)}$ denote the t^{th} sample from $\pi(\boldsymbol{z}, \boldsymbol{\rho}, \boldsymbol{\phi} \mid X, \boldsymbol{y})$ and T denotes the total number of samples drawn.

[†] A GP prior over the function space $z(\cdot)$ implies that the joint distribution of $\boldsymbol{z}$ and $\boldsymbol{z}^*$ has an $(N + N^*)$-dimensional Gaussian distribution. Furthermore, by the properties of Gaussian distribution, the conditional distribution of $\boldsymbol{z}^*$ given $\boldsymbol{z}$ is Gaussian as well (see Appendix A).

Finally, if we are interested in getting "noisy" predictions, $\boldsymbol{y}^* = (y_1^*, \ldots, y_{N^*}^*)$, instead of predictions of the latent function $\boldsymbol{z}^*$, we additionally need to marginalise the latent function values of the new inputs X^*. For computational purposes we typically only consider point-wise predictions at each $\boldsymbol{x}^*$. Thus, we obtain the posterior predictive distribution for a single y^* by

$$\pi(y^* \mid X, \boldsymbol{y}, \boldsymbol{x}^*) = \int p(y^* \mid z^*, \boldsymbol{\rho})\pi(z^* \mid \boldsymbol{z}, X, \boldsymbol{x}^*, \boldsymbol{\phi})\pi(\boldsymbol{z}, \boldsymbol{\rho}, \boldsymbol{\phi} \mid X, \boldsymbol{y})\, dz^*\, d\boldsymbol{z}\, d\boldsymbol{\rho}\, d\boldsymbol{\phi}, \tag{2.10}$$

where, for brevity, we denote z^* the value of the latent function evaluated at $\boldsymbol{x}^*$, i.e $z^* \equiv z(\boldsymbol{x}^*)$. This integral, similar to Eq. (2.7), can be approximated through Monte Carlo integration.

In the next section, we start by focusing our analysis on GP models with Gaussian likelihoods. We discuss how to make inference when the likelihood is non-Gaussian in Section 2.1.1.2 (see Rasmussen and Williams (2006) for an expanded full treatment).

2.1.1.1 Gaussian likelihoods

When the likelihood function is normal, the GP is an appeling prior because the posterior quantities of interest have analytical solutions. The most common example of this is the Gaussian process regression (GPR) model with Gaussian noise. Under this model, the response variable, y_n, is assumed to be a corrupted version of the true latent function $z(\boldsymbol{x}_n)$. More precisely,

$$y_n = z(\boldsymbol{x}_n) + \varepsilon_n, \quad \varepsilon_n \mid \sigma_\varepsilon^2 \overset{\text{iid}}{\sim} \mathrm{N}(0, \sigma_\varepsilon^2), \tag{2.11}$$

where $z : \mathbb{R}^D \to \mathbb{R}$ is the latent function and ε_n is the observation error, which is assumed to be independent and identically normally distributed with noise variance σ_ε^2. Consequently, the corresponding likelihood function is a multivariate Gaussian distribution, $\mathrm{N}(\boldsymbol{z}, \sigma_\varepsilon^2 I_N)$, where I_N denotes the $(N \times N)$ identity matrix.

When making inference about the unknowns in the model, the integral in Eq. (2.4) can be solved analytically, because the prior over the latent functions is conjugate to the likelihood. Specifically, the marginal likelihood is

$$p(\boldsymbol{y} \mid X, \sigma_\varepsilon^2, \boldsymbol{\phi}) = \mathrm{N}(\boldsymbol{\mu}, C_{\boldsymbol{\phi}} + \sigma_\varepsilon^2 I_N), \tag{2.12}$$

and consequently the conditional posterior of the latent function (Eq. (2.4)) is

$$\pi(\boldsymbol{z} \mid X, \boldsymbol{y}, \sigma_\varepsilon^2, \boldsymbol{\phi}) = \mathrm{N}(\boldsymbol{\mu} + C_{\boldsymbol{\phi}}(C_{\boldsymbol{\phi}} + \sigma_\varepsilon^2 I_N)^{-1}(\boldsymbol{y} - \boldsymbol{\mu}), C_{\boldsymbol{\phi}} - C_{\boldsymbol{\phi}}(C_{\boldsymbol{\phi}} + \sigma_\varepsilon^2 I_N)^{-1} C_{\boldsymbol{\phi}}). \tag{2.13}$$

Additionally, the conditional posterior predictive distribution, $\pi(\boldsymbol{z}^* \mid X, \boldsymbol{y}, X^*, \sigma_\varepsilon^2, \boldsymbol{\phi})$[†], is also a multivariate Gaussian with mean and variance given by

$$\begin{aligned} \mathbb{E}\left[\boldsymbol{z}^* \mid X, \boldsymbol{y}, X^*, \sigma_\varepsilon^2, \boldsymbol{\phi}\right] &= \boldsymbol{\mu} + C_{\boldsymbol{\phi}}^* \left(C_{\boldsymbol{\phi}} + \sigma_\varepsilon^2 I_N\right)^{-1} (\boldsymbol{y} - \boldsymbol{\mu}) \\ \mathrm{Cov}\left[\boldsymbol{z}^* \mid X, \boldsymbol{y}, X^*, \sigma_\varepsilon^2, \boldsymbol{\phi}\right] &= C_{\boldsymbol{\phi}}^{**} - C_{\boldsymbol{\phi}}^* \left(C_{\boldsymbol{\phi}} + \sigma_\varepsilon^2 I_N\right)^{-1} C_{\boldsymbol{\phi}}^{*T}. \end{aligned} \tag{2.14}$$

Notice that in Eq. (2.14), the predictive mean is a linear combination of the responses y_n, whereas the predictive covariance does not depend on them. Hence, the conditional posterior distribution of the latent function given the hyperparameters in the model is also a GP with mean and covariance function given by Eq. (2.13).

As mentioned before, a common practice when making predictions is to plug-in point estimates of the parameters and hyperparameters in Eq. (2.14) to approximate $\pi(\boldsymbol{z}^* \mid X, \boldsymbol{y}, X^*)$. When marginalisation is performed through Monte Carlo integration, one can draw T samples from $\pi(\boldsymbol{\rho}, \boldsymbol{\phi} \mid X, \boldsymbol{y}) \propto p(\boldsymbol{y} \mid X, \sigma_\varepsilon^2, \boldsymbol{\phi})\pi(\sigma_\varepsilon^2)\pi(\boldsymbol{\phi})$ and calculate,

$$\mathbb{E}\left[\boldsymbol{z}^* \mid X, \boldsymbol{y}, X^*\right] \approx \frac{1}{T} \sum_{t=1}^{T} \mathbb{E}\left[\boldsymbol{z}^* \mid X, \boldsymbol{y}, X^*, \sigma_\varepsilon^{2^{(t)}}, \boldsymbol{\phi}^{(t)}\right], \tag{2.15}$$

$$\begin{aligned} \mathrm{Cov}\left[\boldsymbol{z}^* \mid X, \boldsymbol{y}, X^*\right] = \mathbb{E}\left[\mathrm{Cov}\left[\boldsymbol{z}^* \mid X, \boldsymbol{y}, X^*, \sigma_\varepsilon^{2^{(t)}}, \boldsymbol{\phi}^{(t)}\right]\right] + \\ \mathrm{Cov}\left[\mathbb{E}\left[\boldsymbol{z}^* \mid X, \boldsymbol{y}, X^*, \sigma_\varepsilon^{2^{(t)}}, \boldsymbol{\phi}^{(t)}\right]\right], \end{aligned} \tag{2.16}$$

where $\sigma_\varepsilon^{2^{(t)}}$ and $\boldsymbol{\phi}^{(t)}$ denote the t^{th} sample from the marginal posterior distribution. Note that the predictive distribution $\pi(\boldsymbol{z}^* \mid X, \boldsymbol{y}, X^*)$ is not Gaussian, but one can evaluate the density throught Monte Carlo approximation. Similarly, to obtain predictions of y^*, we first compute

$$\pi(y^* \mid X, \boldsymbol{y}, \sigma_\varepsilon^2, \boldsymbol{\phi}, \boldsymbol{x}^*) = \int p(y^* \mid z^*, \sigma_\varepsilon^2)\pi(z^* \mid X, \boldsymbol{y}, \boldsymbol{x}^*, \sigma_\varepsilon^2, \boldsymbol{\phi})\, dz^*,$$

which is a Gaussian distribution, with mean as in Eq. (2.14) and covariance given by $\mathrm{Cov}\left[\boldsymbol{z}^* \mid X, \boldsymbol{y}, X^*, \sigma_\varepsilon^2, \boldsymbol{\phi}\right] + \sigma_\varepsilon^2$. Finally, marginalisation over the hyperparameters can be done as in Eq. (2.15) and (2.16).

[†] Notice that $\pi(\boldsymbol{z}^* \mid X, \boldsymbol{y}, X^*, \sigma_\varepsilon^2, \boldsymbol{\phi}) = \int \pi(\boldsymbol{z}^* \mid \boldsymbol{z}, X, X^*, \sigma_\varepsilon^2, \boldsymbol{\phi})\pi(\boldsymbol{z} \mid X, \boldsymbol{y}, \sigma_\varepsilon^2, \boldsymbol{\phi})\, d\boldsymbol{z}$, where the first term in the integral corresponds to Eq. (2.8), and the second term to Eq. (2.13).

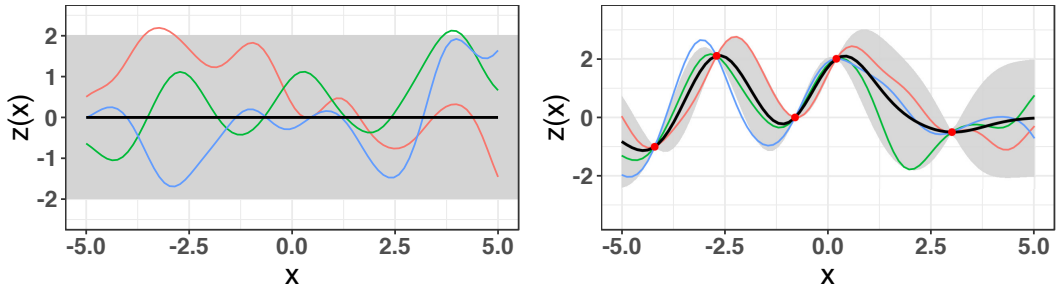


Figure 2.1: Draws from GP prior and posterior with a squared exponential kernel and fixed hyperparameters values (length-scale $\lambda = 0.5$, and magnitude $\tau^2 = 2$). The mean function is denoted in black. The grey area depicts the 95% credible intervals. (Left:) Three draws from a GP prior. (Right:) Three draws from the GP posterior with red dots denoting the observed data.

Figure 2.1 shows an illustration of a GPR model, where we assume noise free observations, such that $y_n = z(\boldsymbol{x}_n)$. We depict draws from a GP prior and posterior. Notice how the observed data restricts the possible functions. Moreover, the uncertainty of the function is higher when we are far from the observed data.

2.1.1.2 Non-Gaussian likelihoods

In the previous section, we explored how to make inference when the likelihood is normal. Under that assumption, conjugacy of the GP prior offers the advantage that the conditional posterior of the latent function has an analytic expression. However, in other applications of GP models, a normality assumption of the data may be inappropriate. An example of this is classification (Rasmussen and Williams, 2006, Chapter 3), or regression problems where the response variable represents proportions or counts (e.g. Vanhatalo and Vehtari, 2007). When the likelihood is not Gaussian, the marginal likelihood in Eq. (2.12) is intractable, and therefore the conditional posterior distribution over the latent function is also intractable. There are several approaches in the literature to overcome this difficulty, which can be deterministic or stochastic.

On one hand, deterministic approximations of the posterior over $\boldsymbol{z}$ include variational methods (Blei et al., 2017), expectation propagation (EP) (Minka, 2001), Laplace approximation (LA) (Kimeldorf and Wahba, 1970; Williams and Barber, 1998), and integrated nested Laplace approximation (INLA) (Rue et al., 2009). Firstly, variational methods minimise the Kullback-Leibler (KL) divergence between an approximate and the exact posterior of interest. The approximations offered by

variational methods may be restricted to be Gaussian (e.g. Williams and Barber, 1998; Opper and Archambeau, 2009) or assumed to have a more general form (Hensman et al., 2015). Secondly, EP approximates each component of the likelihood function (which must factorise over the data points) with a distribution restricted to be in the exponential family, that is chosen to minimise the KL divergence between the true and the approximated distributions. These local approximations in turn provide an approximation of the posterior of $\boldsymbol{z}$. Thirdly, LA employs a Gaussian approximation of $\pi(\boldsymbol{z} \mid X, \boldsymbol{y}, \boldsymbol{\rho}, \boldsymbol{\phi})$ (Eq. (2.4)), such approximated Gaussian is obtained through a second-order Taylor expansion of the log posterior. Kuss and Rasmussen (2006) compared the performance of EP and LA for binary classification problems, concluding that the former method outperforms LA, and warned of high inaccuracies of LA in this setting. Finally, INLA exploits a Gaussian Markov representation of the process and uses a LA for each component in the conditional posterior of $\boldsymbol{z}$, i.e $\pi(z_n \mid X, \boldsymbol{y}, \boldsymbol{\phi}, \boldsymbol{\rho})$, and for the marginal posterior, $\pi(\boldsymbol{\phi}, \boldsymbol{\rho} \mid \boldsymbol{y}, X)$. These approximated densities are later employed to obtain marginal posteriors of $\pi(z_n \mid X, \boldsymbol{y})$ by numerically integrating out $\boldsymbol{\phi}$ and $\boldsymbol{\rho}$. INLA offers computational advantages by working with sparse matrices.

On the other hand, stochastic approximations employ MCMC techniques to obtain samples from the joint posterior distribution of the latent variables and the hyperparameters, $\pi(\boldsymbol{z}, \boldsymbol{\rho}, \boldsymbol{\phi} \mid X, \boldsymbol{y})$. However, it is well known in the literature (Vanhatalo and Vehtari, 2007; Rue et al., 2009; Filippone and Girolami, 2014) that efficient sampling of both latent function and hyperparameters is challenging because they tend to be highly correlated. When employing a Gibbs scheme, sampling from $\pi(\boldsymbol{z} \mid X, \boldsymbol{y}, \boldsymbol{\rho}, \boldsymbol{\phi})$ can be done with elliptical slice sampling (Ell-SS) (Murray et al., 2010), the preconditioned Crank–Nicolson (pCN) (Beskos et al., 2008) sampler or its Langevin variant (Cotter et al., 2013), a Metropolis-adjusted Langevin algorithm (MALA) (Roberts et al., 1996), Hamiltonian Monte Carlo (HMC) (Duane et al., 1987) or with the recently proposed auxiliary gradient-based schemes of Titsias and Papaspiliopoulos (2018). Sampling from $\pi(\boldsymbol{\rho}, \boldsymbol{\phi} \mid X, \boldsymbol{y}, \boldsymbol{z})$ is typically done with a random walk Metropolis-Hastings (RW-MH) algorithm (possibly adaptive), but MALA, HMC or other method can be used. To break the strong correlation between $\boldsymbol{z}$ and $\boldsymbol{\phi}$ one can employ the whitening approach discussed by Murray and Adams (2010), the interweaving parametrisation of Yu and Meng (2011) or a pseudo-marginal scheme (Filippone and Girolami, 2014; Xiong et al., 2017).

2.1.2 The covariance function

Both the spatial statistics and the machine learning literature have suggested several forms of covariance functions and discussed their properties. The most common kernels are presented in this section; however, it is worth mentioning that the covariance functions shown here are just some of the available in closed-form. The literature is vast in this respect and other possibilities are available. A more complete overview is provided in Finkenstadt et al. (2006); Rasmussen and Williams (2006); and Sherman (2011). Furthermore, it is possible to create new covariance functions from already existing kernels; for instance by adding, multiplying, convolving or combining valid kernels (Section 4.2.4, Rasmussen and Williams, 2006; Álvarez et al., 2012; Yang et al., 2016).

2.1.2.1 Stationary covariance functions

A GP with a constant mean and stationary covariance function is called a *stationary* or *homogeneous Gaussian process* (Abrahamsen, 1997). A covariance function is said to be stationary if it is invariant under translations, i.e. $C(\boldsymbol{x}_i, \boldsymbol{x}_j) = C(\boldsymbol{x}_i + \delta, \boldsymbol{x}_j + \delta)$, for any $\delta \in \mathcal{X}$. Therefore, stationary kernels are function of the separation vector $\boldsymbol{x}_i - \boldsymbol{x}_j$. If the covariance function is also invariant to rotations, then it is said to be *isotropic*. Isotropic functions depend only on the Euclidean norm, $\|\cdot\|$, of the input such that $C(\boldsymbol{x}_i, \boldsymbol{x}_j) = C(\|\boldsymbol{x}_i - \boldsymbol{x}_j\|)$. In order to clarify the notation, we distinguish between stationary, C^{S}, and non-stationary, C^{NS}, covariance functions, when required.

Among stationary kernels, the use of the Matérn covariance family is widespread in spatial statistics, being the default choice in several applications (Gelfand et al., 2010). Various parametrisations of this covariance function are available in the literature (e.g. Lindgren et al., 2011; Finkenstadt et al., 2006; Gelfand et al., 2010; Rasmussen and Williams, 2006). Here, we present the parametrisation in Rasmussen and Williams (2006); therefore, the stationary isotropic Matérn kernel is

$$C^{\mathrm{S}}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \frac{\tau^2 2^{1-\nu}}{\Gamma(\nu)} \left(\frac{\sqrt{2\nu}\|\boldsymbol{x}_i - \boldsymbol{x}_j\|}{\lambda} \right)^{\nu} \mathcal{K}_\nu \left(\frac{\sqrt{2\nu}\|\boldsymbol{x}_i - \boldsymbol{x}_j\|}{\lambda} \right), \tag{2.17}$$

where $\Gamma(\cdot)$ is the gamma-function, $\nu, \tau^2, \lambda > 0$, and $\mathcal{K}_\nu(\cdot)$ denotes the modified Bessel function of the second kind of order ν. Other common covariance functions arise as special cases of the Matérn family. In particular, when $\nu = 1/2$ the stationary

isotropic Matérn family is reduced to the isotropic exponential kernel,

$$C^{\mathrm{S}}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \tau^2 \exp\left(-\frac{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|}{\lambda}\right), \tag{2.18}$$

and when $\nu \to \infty$ the Matérn class gives rise to the isotropic squared exponential (SE),

$$C^{\mathrm{S}}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \tau^2 \exp\left(-\frac{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2}{2\lambda^2}\right), \tag{2.19}$$

which is also referred to as the radial basis function (RBS) kernel and is frequently employed in the machine learning literature.

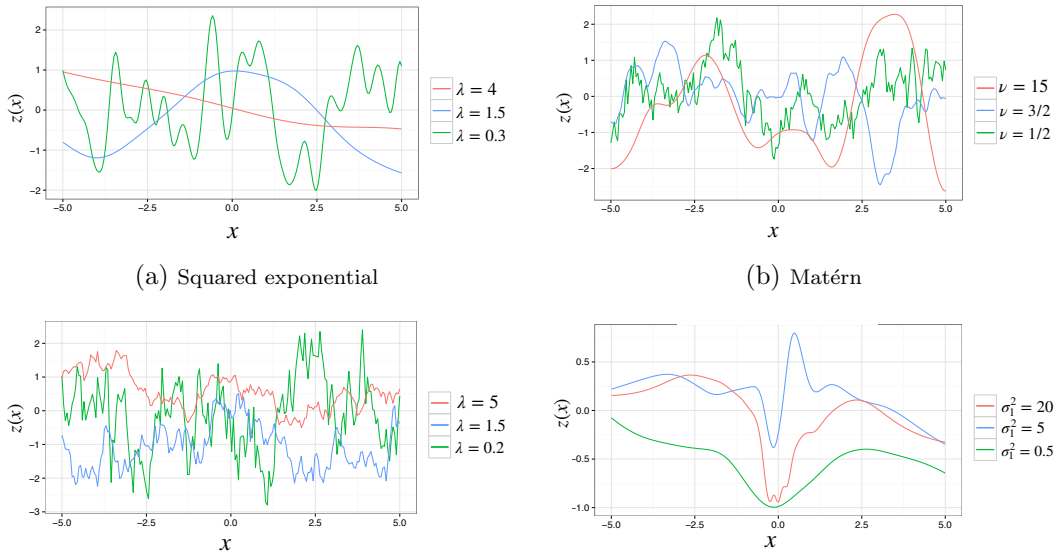
An *anisotropic* covariance function is a stationary function that depends on $\boldsymbol{x}_i - \boldsymbol{x}_j$ through a non-Euclidean norm (Abrahamsen, 1997). Therefore, an anisotropic covariance can be constructed from an isotropic one by replacing the Euclidean norm with a Mahalanobis norm; namely, $\sqrt{(\boldsymbol{x}_i - \boldsymbol{x}_j)^T \Lambda (\boldsymbol{x}_i - \boldsymbol{x}_j)}$, where Λ is a positive semi-definite matrix. A common choice is $\Lambda = \mathrm{diag}(\lambda_1^{-2}, \ldots, \lambda_D^{-2})$, which assumes a different length-scale parameter in each dimension and results in an automatic relevance determination (ARD) kernel. Note that in the case when a spherical structure in $\Lambda = \lambda^{-2} I_D$ is assumed, we recover an isotopic formulation. Some other possible choices for Λ are discussed in Rasmussen and Williams (2006, Chapter 5).

Notice the parametric form of Eq. (2.17)-(2.19), which is a common characteristic of closed-form covariance functions. In general, the parameters in the covariance function control its properties and therefore the realisations of the process. Below we discuss the impact of the hyperparameters.

First, λ denotes the length-scale parameter, also referred to as the range, which controls how the correlation collapses with distance (Finkenstadt et al., 2006). Large values of λ characterise highly correlated observations, describing functions that change slowly. In addition, the length-scale parameter plays a vital role in ARD kernels to perform feature selection (MacKay, 1996); the magnitude of λ_d will indicate the negligibly of the d^{th} input dimension. Second, the magnitude parameter, denoted by τ^2, describes the variance of the process and measures how far the function is from its mean. The magnitude is also called the signal variance and can be seen as a scaling factor which makes a correlation function a covariance function, such that any kernel function $C(\boldsymbol{x}_i, \boldsymbol{x}_j) = \tau^2 R(\boldsymbol{x}_i, \boldsymbol{x}_j)$, where $R(\boldsymbol{x}_i, \boldsymbol{x}_j)$ is a correlation function. Third, the Matérn family has a differentiability parameter, denoted by ν. This parameter is of central importance when interpolating (Finkenstadt et al., 2006) because it provides flexibility in the local behaviour of the process (Stein,

2012). The bigger the value of ν, the smoother the process, see Figure 2.2(b) for an illustration. Recall that when $\nu = 1/2$, the stationary Matérn family is reduced to the exponential kernel, a covariance function that generates rough sample paths (see Figure 2.2(c)); specifically, for $D = 1$, this choice results in an autoregressive process with lag one. In contrast, as $\nu \to \infty$, we recover the SE covariance, a kernel which realisations are infinitely differentiable, producing very smooth sample functions as depicted in Figure 2.2(a).

A large amount of research and applications has focused on stationary models due to ease in implementation. However, stationarity, as a modelling assumption, may not be accurate in situations where the correlation is not constant along the input space. In such cases, the correlation of the process is spatial or input dependent. As a consequence, several efforts have been made to consider non-stationary processes, which allow more flexible and appropriate models for problems that arise in various applications, such as those that occur in epidemiology or environmental sciences.



(a) Squared exponential

(b) Matérn

(c) Exponential

(d) Neural network

Figure 2.2: The effect of varying the hyperparameters in the covariance function. (a): Varying length-scale parameter, λ, in the squared exponential kernel with magnitude fixed at $\tau^2 = 1$. (b): Varying the smoothness parameter, ν, in the Matérn class with $\tau^2 = 1$ and $\lambda = 1$ fixed. (c): Varying λ in the exponential kernel, with fixed $\tau^2 = 1$. (d): Varying σ_1^2 in the neural network kernel with fixed $\sigma_0^2 = 1$.

2.1.2.2 Non-stationary covariance functions

A GP is said to be *non-stationary* if it is not invariant with respect to translations. Recent reviews on modelling approaches to deal with non-stationarity are provided by Fouedjio (2016) and Volodina and Williamson (2018). Some of the available methodologies are deformation techniques (Sampson et al., 2001; Schmidt and O'Hagan, 2003; Anderes and Stein, 2008); partition methods (Kim et al., 2005; Gramacy and Lee, 2012); basis functions representation (Holland et al., 1999; Nychka et al., 2002; Stephenson et al., 2005); the stochastic partial differential equation approach of Lindgren et al. (2011); and process convolution methods introduced by Higdon (1998) and later employed by Paciorek (2003).

A non-stationary covariance function allows the process to depend not only on the separation vector but also on the location. Non-stationary kernels offer more flexibility, introducing spatial dependence, which makes them more appropriate in a wide variety of applications. For example, they are required in examples when the function is very flat in some regions and changes more rapidly in others. Some closed-forms non-stationary kernels include the neural network (Neal, 1995; Williams, 1998), the Gibbs kernel (Gibbs, 1997), and non-stationary versions of the Matérn class (Paciorek, 2003; Paciorek and Schervish, 2006)

Firstly, the neural network kernel is

$$C^{\text{NS}}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \frac{2\tau^2}{\pi} \arcsin\left(\frac{2\breve{\boldsymbol{x}}_i^T \Upsilon \breve{\boldsymbol{x}}_j}{\sqrt{(1 + 2\breve{\boldsymbol{x}}_i^T \Upsilon \breve{\boldsymbol{x}}_i)(1 + 2\breve{\boldsymbol{x}}_j^T \Upsilon \breve{\boldsymbol{x}}_j)}} \right), \tag{2.20}$$

where $\breve{\boldsymbol{x}}_i = (1, x_{i1}, x_{i2}, ..., x_{iD})^T$ is an augmented version of the vector $\boldsymbol{x}_i$, and Υ is a $(D+1) \times (D+1)$ diagonal matrix, $\Upsilon = \text{diag}(\sigma_0^2, \sigma_1^2, \sigma_2^2, ..., \sigma_D^2)$, which contains variance parameters. The first entry, σ_0^2, is the variance of the bias, and the remaining diagonal entries correspond to variance parameters that control the scaling along each dimension of $\boldsymbol{x}$ (Rasmussen and Williams, 2006). As with other kernels, τ^2 denotes the magnitude parameter. This kernel produces more flexible functions as the values of the variances $\sigma_1^2, \sigma_2^2, ..., \sigma_D^2$ increase (see Figure 2.2(d)). More details on how this kernel function is derived are available in Williams (1998).

Secondly, Paciorek and Schervish (2006) introduced a family of non-stationary covariance functions, which generalised the Gibbs kernel (Gibbs, 1997). The result derived by Paciorek and Schervish (2006) states that for any stationary, isotropic correlation function, R_{ψ}, which is positive definite on $\mathbb{R}$ and depends on parameters

$\boldsymbol{\psi}$, we can obtain a non-stationary covariance function through

$$C^{\text{NS}}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \frac{\tau^2 |\Sigma(\boldsymbol{x}_i)|^{\frac{1}{4}} \left|\Sigma(\boldsymbol{x}_j)\right|^{\frac{1}{4}}}{\left|\frac{\Sigma(\boldsymbol{x}_i)+\Sigma(\boldsymbol{x}_j)}{2}\right|^{\frac{1}{2}}} R_{\boldsymbol{\psi}}\left(\sqrt{G_{ij}}\right), \tag{2.21}$$

where

$$G_{ij} = \left(\boldsymbol{x}_i - \boldsymbol{x}_j\right)^T \left(\frac{\Sigma(\boldsymbol{x}_i) + \Sigma(\boldsymbol{x}_j)}{2}\right)^{-1} \left(\boldsymbol{x}_i - \boldsymbol{x}_j\right).$$

In the above formulation, $\Sigma(\boldsymbol{x}_i)$ and $\Sigma(\boldsymbol{x}_j)$ are covariance matrices at locations $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ respectively. These covariance matrices, referred to as kernel matrices by Paciorek and Schervish (2006), are key components to introduce non-stationarity in the correlation function and are added to the formula through the convolution of two (multivariate) Gaussian densities centred at each location (details on the derivations are provided in Paciorek (2003, Section 2.2)). Furthermore, Paciorek (2003, Chapter 2) proved that the smoothness properties of the non-stationary covariance function in Eq. (2.21) are inherited from the properties of the stationary correlation function, $R_{\boldsymbol{\psi}}$, employed to construct it, as long as the covariance matrices at each location vary smoothly.

Of particular interest is the non-stationary version of the Matérn family,

$$C^{\text{NS}}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \frac{\tau^2 2^{1-\nu} |\Sigma(\boldsymbol{x}_i)|^{\frac{1}{4}} \left|\Sigma(\boldsymbol{x}_j)\right|^{\frac{1}{4}}}{\Gamma(\nu)\left|\frac{\Sigma(\boldsymbol{x}_i)+\Sigma(\boldsymbol{x}_j)}{2}\right|^{\frac{1}{2}}} \left(\sqrt{2\nu G_{ij}}\right)^{\nu} \mathcal{K}_{\nu}\left(\sqrt{2\nu G_{ij}}\right). \tag{2.22}$$

As in the stationary Matérn kernel, ν is the smoothness parameter and τ^2 is the variance. Notice that, except for the kernel matrices, the interpretation of the parameters in the non-stationary Matérn kernel is the same as in its stationary version. We present more details about the kernel matrices hyperparameters in Section 2.3.

Finally, we present the non-stationary squared exponential, also derived from Eq. (2.21),

$$C^{\text{NS}}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \frac{\tau^2 |\Sigma(\boldsymbol{x}_i)|^{\frac{1}{4}} \left|\Sigma(\boldsymbol{x}_j)\right|^{\frac{1}{4}}}{\left|\frac{\Sigma(\boldsymbol{x}_i)+\Sigma(\boldsymbol{x}_j)}{2}\right|^{\frac{1}{2}}} \exp\left(-\frac{G_{ij}}{2}\right). \tag{2.23}$$

When the kernel matrices are assumed to be diagonal, this formulation corresponds to the Gibbs covariance function.

2.1.2.3 Separable covariance functions

Finally, either stationary or non-stationary kernels can be categorised as *separable* or *non-separable*. A covariance function is said to be *separable* if $C(\boldsymbol{x}_i, \boldsymbol{x}_j) = \prod_{d=1}^{D} C(x_{id}, x_{jd})$. More precisely, the covariance is called *partially separable* when $C(\boldsymbol{x}_i, \boldsymbol{x}_j) = \left(\prod_{p=1}^{P} C(x_{ip}, x_{jp})\right) C(\boldsymbol{x}'_i, \boldsymbol{x}'_j)$ with $\boldsymbol{x}' \in \mathbb{R}^{D-P}$ for $P < D$, and *fully separable* when $P = D$ (Abrahamsen, 1997). Separable covariance functions are widely employed to model spatio-temporal data, by dividing the covariance function into two components, one for the spatial element and the second one for the time component. Even though separability offers computational advantages (by reducing the computational cost of the required matrix inversion), it can also be restrictive, by not allowing interactions between dimensions (Genton, 2007).

2.2 Overview on MCMC methods

Markov chain Monte Carlo methods are a family of algorithms employed to sample from arbitrary distributions. In Bayesian inference the implementation of MCMC techniques has allowed the development of complex models, by permitting sampling from complicated posterior distributions, that can be known up to a proportionality constant. These methods generate a correlated chain, which under certain conditions, will converge to the distribution of interest (the stationary or target distribution) (Robert and Casella, 2013).

In this section, we give a short and concise presentation of the MCMC methods we employ throughout this thesis. Some useful references for a more detail review are Gilks et al. (1995); Robert and Casella (2013); Gelman et al. (2014) and Brooks et al. (2011). Let us assume that we aim to sample from the posterior distribution $\pi(\boldsymbol{\theta})$ of a K-dimensional parameter $\boldsymbol{\theta}$.

2.2.1 Metropolis-Hastings algorithm

First, we present the Metropolis-Hastings (MH) algorithm (Metropolis et al., 1953; Hastings, 1970). The idea behind this sampler is that, given a density $g(\cdot)$ that is easy to sample from, we can propose values that will be accepted or rejected as samples coming from the target density $\pi(\boldsymbol{\theta})$ (Robert and Casella, 2013).

Algorithm 1 Metropolis-Hastings (MH)

Require: Target distribution: π, current state $\boldsymbol{\theta}^{(t)}$, and proposal distribution g

1: **procedure** MH($\boldsymbol{\theta}^{(t)}, g$)

2: Draw $\boldsymbol{\theta}' \sim g(\cdot \mid \boldsymbol{\theta}^{(t)})$

3: Compute acceptance probability:

$$c \leftarrow \min\left\{1, \frac{\pi(\boldsymbol{\theta}')g(\boldsymbol{\theta}^{(t)} \mid \boldsymbol{\theta}')}{\pi(\boldsymbol{\theta}^{(t)})g(\boldsymbol{\theta}' \mid \boldsymbol{\theta}^{(t)})}\right\}$$

4: **if** Unif$[0,1] < c$ **then** $\boldsymbol{\theta}^{(t+1)} \leftarrow \boldsymbol{\theta}'$ **else** $\boldsymbol{\theta}^{(t+1)} \leftarrow \boldsymbol{\theta}^{(t)}$ **end if**

5: **return** $\boldsymbol{\theta}^{(t+1)}$

6: **end procedure**

When the selected proposal distribution is symmetric, the acceptance probability in Step 3 of Algorithm 1 is simplified. This is the case for the popular random walk Metropolis-Hastings (RW-MH) algorithm (Hastings, 1970), detailed in Algorithm 2, where the instrumental density g proposes values $\boldsymbol{\theta}' \sim \mathrm{N}(\boldsymbol{\theta}^{(t)}, \Sigma)$.

Algorithm 2 Random walk Metropolis-Hastings (RW-MH)

Require: Target distribution: π, current state $\boldsymbol{\theta}^{(t)}$, and proposal variance Σ

1: **procedure** RW-MH($\boldsymbol{\theta}^{(t)}, \Sigma$)

2: Draw $\boldsymbol{\theta}' \sim \mathrm{N}(\boldsymbol{\theta}^{(t)}, \Sigma)$

3: Compute acceptance probability:

$$c \leftarrow \min\left\{1, \frac{\pi(\boldsymbol{\theta}')}{\pi(\boldsymbol{\theta}^{(t)})}\right\}$$

4: **if** Unif$[0,1] < c$ **then** $\boldsymbol{\theta}^{(t+1)} \leftarrow \boldsymbol{\theta}'$ **else** $\boldsymbol{\theta}^{(t+1)} \leftarrow \boldsymbol{\theta}^{(t)}$ **end if**

5: **return** $\boldsymbol{\theta}^{(t+1)}$

6: **end procedure**

Acceptance rates in MH algorithms are a crucial component, as they are related to mixing and convergence properties of the chains. A complete discussion of this topic is provided by Robert and Casella (2009). Here, we limit our exposition by summarizing that higher acceptance rates do not necessarily produce good explo-

ration of the space and fast mixing of the chains. In particular, for the random walk MH in Algorithm 2, neither the highest acceptance rate nor the lowest leads to the best mixing (Robert and Casella, 2009). An *optimal* acceptance rate, suggested by Roberts et al. (1997), is 0.234 in multidimensional settings and 0.44 for dimension one or two. Precisely, the scale of the instrumental density in Algorithm 2 plays a crucial role to achieve this *optimal* value. Therefore, the performance of the algorithm can be improved by tuning the scale parameter of the proposal distribution. However, performing such a task manually can be time-consuming and inefficient. Consequently, adaptive schemes that automatically tune the scale parameter have been proposed. Some useful references for adaptive MCMC algorithms include Roberts and Rosenthal (2009), Liang et al. (2011, Chapter 8), and Damien et al. (2013, Chapter 7)

2.2.2 Gibbs sampler

Second, a special case of the MH algorithm, in which the acceptance probability is always one, is the Gibbs sampler (Geman and Geman, 1987), which is illustrated in Algorithm 3. This scheme breaks the posterior of interest $\pi(\boldsymbol{\theta})$, into the full conditional distributions, that we denote $\pi_{\theta_k|\boldsymbol{\theta}_{-k}}$. The notation $\boldsymbol{\theta}_{-k}$ indicates the vector $\boldsymbol{\theta}$ with the k^{th} component removed. When the full conditionals are available in closed-form, the Gibbs sampler is particularly convenient. However, when this is not the case, may not be easy to sample from. In such a situation, it is possible to combine Algorithm 1 with Algorithm 3 resulting in the Metropolis-within-Gibbs (MWG) sampler, where we include an MH step for each of these intractable distributions. Moreover, when MWG is combined with adaptation, this is usually referred to as an adaptive Metropolis-within-Gibbs sampler.

The scheme that we illustrate in Algorithm 3 is the systematic scan sampler with parameter by parameter updates. However, it is worth mentioning that there exist variations of this algorithm. For instance, it is possible to group components of $\boldsymbol{\theta}$ and sample them in blocks from their full conditionals, resulting in a block Gibbs sampler. Also, the updated component can be selected at random rather than in a preferred order, obtaining the random scan Gibbs sampler.

2.2.3 Elliptical slice sampling

Building upon slice sampling (Neal et al., 2003), Murray et al. (2010) proposed elliptical slice sampling (Ell-SS), an MCMC method to sample from posterior distributions with multivariate Gaussian priors. The scheme assumes that the posterior

Algorithm 3 Systematic scan Gibbs (Gibbs)

Require: Target distribution: π, current state $\boldsymbol{\theta}^{(t)}$, and full conditionals $\pi_{\theta_k|\boldsymbol{\theta}_{-k}}$, for $k = 1, \ldots, K$

procedure Gibbs $(\boldsymbol{\theta}^{(t)}, \pi_{\theta_1|\boldsymbol{\theta}_{-1}}, \ldots \pi_{\theta_K|\boldsymbol{\theta}_{-K}})$

1: Draw $\theta_1^{(t+1)} \sim \pi_{\theta_1|\boldsymbol{\theta}_{-1}}(\cdot \mid \theta_2^{(t)}, \ldots \theta_K^{(t)})$

$\vdots$

k: Draw $\theta_k^{(t+1)} \sim \pi_{\theta_k|\boldsymbol{\theta}_{-k}}(\cdot \mid \theta_1^{(t)}, \ldots \theta_{k-1}^{(t+1)}, \theta_{k+1}^{(t)}, \ldots \theta_K^{(t)})$

$\vdots$

K: Draw $\theta_K^{(t+1)} \sim \pi_{\theta_K|\boldsymbol{\theta}_{-K}}(\cdot \mid \theta_1^{(t+1)}, \ldots, \theta_{K-1}^{(t+1)})$

return $\boldsymbol{\theta}^{(t+1)}$

end procedure

of interest has the form $\pi(\boldsymbol{\theta}) \propto \mathcal{L}(\boldsymbol{\theta})\mathrm{N}(\boldsymbol{\theta} \mid 0, \Omega)$, where $\mathcal{L}(\boldsymbol{\theta})$ is the likelihood function, and $\mathrm{N}(\boldsymbol{\theta} \mid 0, \Omega)$ is the multivariate Gaussian prior. Algorithm 4 describes Ell-SS. Importantly, this method has been shown to be successful to sample the

Algorithm 4 Elliptical slice sampling (Ell-SS)

Require: Likelihood function: $\mathcal{L}(\boldsymbol{\theta})$, prior covariance Ω, and current state $\boldsymbol{\theta}^{(t)}$

1: **procedure** Ell-SS$(\boldsymbol{\theta}^{(t)}, \Omega)$
2: $\quad\boldsymbol{\eta} \sim \mathrm{N}(0, \Omega)$
3: $\quad c \sim \mathrm{Unif}[0, 1]$
4: $\quad\kappa \leftarrow \log \mathcal{L}(\boldsymbol{\theta}^{(t)}) + \log c$
5: $\quad\gamma \sim \mathrm{Unif}[0, 2\pi]$
6: $\quad[\gamma_{\min}, \gamma_{\max}] \leftarrow [\gamma - 2\pi, \gamma]$
7: $\quad$**repeat**
8: $\quad\quad\boldsymbol{\theta}^{(t+1)} \leftarrow \boldsymbol{\theta}^{(t)} \cos\gamma + \boldsymbol{\eta} \sin\gamma$
9: $\quad\quad$**if** $\gamma < 0$ **then** $\gamma_{\min} \leftarrow \gamma$ **else** $\gamma_{\max} \leftarrow \gamma$ **end if**
10: $\quad\quad\gamma \sim \mathrm{Unif}[\gamma_{\min}, \gamma_{\max}]$
11: $\quad$**until** $\log \mathcal{L}(\boldsymbol{\theta}^{(t+1)}) > \kappa$
12: $\quad$**return** $\boldsymbol{\theta}^{(t+1)}$
13: **end procedure**

latent function values in GP models (Murray et al., 2010; Filippone et al., 2013). Ell-SS is a rejection free method, free of parameter tuning, with the advantage of

straightforward implementation.

2.2.4 MCMC for Gaussian process models

The MH algorithm and the Gibbs sampler are the most frequently employed MCMC techniques when performing inference. However, in models with parameters that are strongly coupled, such as in GP models, these methods might perform poorly in terms of mixing and efficiency of the chains. Specifically, efficient sampling from $\pi(\boldsymbol{z}, \boldsymbol{\rho}, \boldsymbol{\phi} \mid X, \boldsymbol{y})$ in GP models is challenging because (i) $\boldsymbol{z}$ is a high-dimensional vector whose elements can be highly correlated and (ii) $\boldsymbol{z}$ and $\boldsymbol{\phi}$ tend to be strongly coupled.

When the latent function can be integrated out (GPR models with Gaussian noise) the hyperparameters can be sampled with a MH step by finding suitable proposal distributions. Depending on the dimension of $\boldsymbol{\phi}$, this can be done jointly or parameter by parameter (Gibbs style). Nonetheless, care must be taken as some covariance parameters can present identifiability issues; see Zhang (2004). For the noise variance, a straightforward approach is to employ a random walk MH step (with adaptation if required) over a transformed parameter, e.g. logarithm or square root.

If the latent function cannot be marginalised, we require draws from the full posterior $\pi(\boldsymbol{z}, \boldsymbol{\rho}, \boldsymbol{\phi} \mid X, \boldsymbol{y})$. To do so, one can use a block Gibbs sampler that alternates between the full conditionals $\pi(\boldsymbol{z} \mid X, \boldsymbol{y}, \boldsymbol{\rho}, \boldsymbol{\phi})$ and $\pi(\boldsymbol{\rho}, \boldsymbol{\phi} \mid X, \boldsymbol{y}, \boldsymbol{z})$. Sampling from $\pi(\boldsymbol{z} \mid X, \boldsymbol{y}, \boldsymbol{\rho}, \boldsymbol{\phi})$ can be done with the Ell-SS scheme described in Algorithm 4; whereas sampling from $\pi(\boldsymbol{\rho}, \boldsymbol{\phi} \mid X, \boldsymbol{y}, \boldsymbol{z})$ can be done with a MH step. Furthermore, because of the high dependence that exists between the latent function and hyperparameters it can be convenient to employ a re-parametrisation to break such correlation. A simple and computationally cheap strategy is to employ the non-centred parametrisation of Papaspiliopoulos et al. (2007). The idea behind this reparametrisation, also referred as ancillary augmentation (Yu and Meng, 2011) or whitened parametrisation, is to define a transformation such that $\boldsymbol{z}$ and $\boldsymbol{\phi}$ are a priori independent. This is achieved through, $\boldsymbol{z} = L(\boldsymbol{\phi})\boldsymbol{\xi} + \boldsymbol{\mu}$, with $L(\boldsymbol{\phi})L(\boldsymbol{\phi})^T = C_{\boldsymbol{\phi}}$ and $\boldsymbol{\xi} \sim \mathrm{N}(0, I_N)$. In this case, we sample $\boldsymbol{\xi}$ rather that $\boldsymbol{z}$ and then sample $\boldsymbol{\phi}$ conditioned on the value of $\boldsymbol{\xi}$. Given sampled values of $\boldsymbol{\phi}$ and $\boldsymbol{\xi}$, we can deterministically obtain $\boldsymbol{z}$. This and other aproaches for breaking the correlation in GPs are discussed by Murray and Adams (2010).

Finally, in addition to the MCMC algorithms discussed here, there exist other approaches to implement Bayesian inference in GP models. For instance, Titsias et al.

(2011) propose a MH scheme that use control variables to sample from an augmented posterior efficiently. Other successful schemes usually incorporate gradient information, such as HMC, Metropolis adjusted Langevin (Roberts and Rosenthal, 1998), Riemann manifold HMC (Girolami and Calderhead, 2011), and auxiliary gradient algorithms proposed in the recent work of Titsias and Papaspiliopoulos (2018).

2.2.5 Further considerations for MCMC implementation

Convergence diagnostics of the chain play an essential role when utilising MCMC methods. Although MCMC algorithms posses theoretical guarantees of convergence of the chain to the stationary distribution, that does not necessarily imply that a sample path will reach stationarity in a limited time frame. Therefore, it is always necessary to monitor the chain before making inference. Several techniques to monitor convergence are available, such as a non-parametric test of stationarity, comparison of multiple chains, and path plots of the chain. More detail on these and other techniques can be found either in Gelman et al. (2014) or Robert and Casella (2013). In this thesis, we use diagnostic plots; namely traceplots and plots of the cumulative averages, and the effective sample size (ESS) as a measure of efficiency of the chains (Brooks et al., 2011). The ESS of T MCMC samples is

$$\text{ESS} = \frac{T}{1 + 2\sum_{s=1}^{\infty} \rho(s)},$$

where $\rho(s)$ denotes the autocorrelation at lag s.

Additionally, two practical considerations to keep in mind are the *burn-in* period and the *thinning* of the chain. The former refers to discarding initial iterations of the chain to minimise the influence of the starting point. The latter helps reduce the natural autocorrelation of the chain by keeping only multiples of the thinning factor.

2.3 Non-stationary multi-level GP models

We introduce now a multi-level GP, which is constructed based on the family of non-stationary covariance functions discussed in Eq. (2.21). This family of closed-form kernels have given rise to different schemes in the literature to model non-stationary datasets. Firstly, Stein (2005) extended the results from Paciorek (2003) and the work of Pintore and Holmes (2004) to obtain an extremely flexible kernel, which corresponds to a generalisation of the non-stationary Matérn in Eq. (2.22)

where all parameters are allowed to vary in space; however, he also pointed out that, even for a fixed ν, spatially varying $\tau^2(\cdot)$ and $\Sigma(\cdot)$ leads to problems of consistent estimation in the parameters. Later, Kleiber and Nychka (2012) developed further the work of Stein (2005) by extending the kernel to multivariate settings, and more recently, Risser and Calder (2015) derived a class of non-stationary kernels that enable the use of covariate information to drive non-stationarity.

In this thesis, we focus on the non-stationary family of kernels derived by Paciorek (2003), where the non-stationarity is introduced by allowing only one of the parameters, namely $\Sigma(\cdot)$, to vary in space. Different approaches to model this spatially varying parameter have been proposed in the literature. For instance, Lang et al. (2007) modelled the kernel matrices, $\Sigma(\cdot)$, through an adaptive procedure and Neto et al. (2014) employed directional covariates to parametrise the kernel matrices.

In particular, we follow the approach of Paciorek (2003) and Higdon et al. (1999), where a second latent GP is employed to model the latent field of spatially varying parameters. This method provides information about how the correlation changes along the input space by analysing the latent field of kernel matrices.

Here, we start by defining a 2-level GP model for one-dimensional problems. In order to perform inference, we use the hierarchical model described in Eq. (2.2). The critical component in the hierarchy is the non-stationary GP prior assigned for the latent function, $z(\cdot)$, which employs the family of non-stationary kernels in Eq. (2.21). We note that for one-dimensional problems, the kernel matrices in the covariance are reduced to scalars, such that $\Sigma(\cdot) := \ell^2(\cdot)$. In addition, the prior for the spatially varying length-scale is assigned over a log-transformed parameter, defined as $u(\cdot) := \log \ell(\cdot)$. The hierarchical formulation of the model is

$$
\begin{aligned}
y_n &\sim p(y_n \mid z(x_n), \boldsymbol{\rho}), \quad n = 1, ..., N, \\
z(\cdot) &\sim \mathrm{GP}\left(\mu_z, C^{\mathrm{NS}}_{\boldsymbol{\phi}}(\cdot,\cdot)\right), \\
\boldsymbol{\rho} &\sim \pi(\boldsymbol{\rho}), \\
u(\cdot) &\sim \mathrm{GP}\left(\mu_u, C^{\mathrm{S}}_{\boldsymbol{\varphi}}(\cdot,\cdot)\right), \\
(\boldsymbol{\psi}, \tau_z^2) &\sim \pi(\boldsymbol{\psi})\pi(\tau_z^2), \\
\boldsymbol{\varphi} &\sim \pi(\boldsymbol{\varphi}),
\end{aligned}
\tag{2.24}
$$

where $\boldsymbol{\phi} = \{\tau_z^2, \boldsymbol{\psi}, u(\cdot)\}$ denotes all the parameters required for the non-stationary kernel, $\boldsymbol{\varphi}$ are parameters of a stationary covariance function, μ_z and μ_u represent the a priori constant mean of the latent function and the log length-scale process, respectively. A plate diagram of this model is given in Figure 2.3(left).

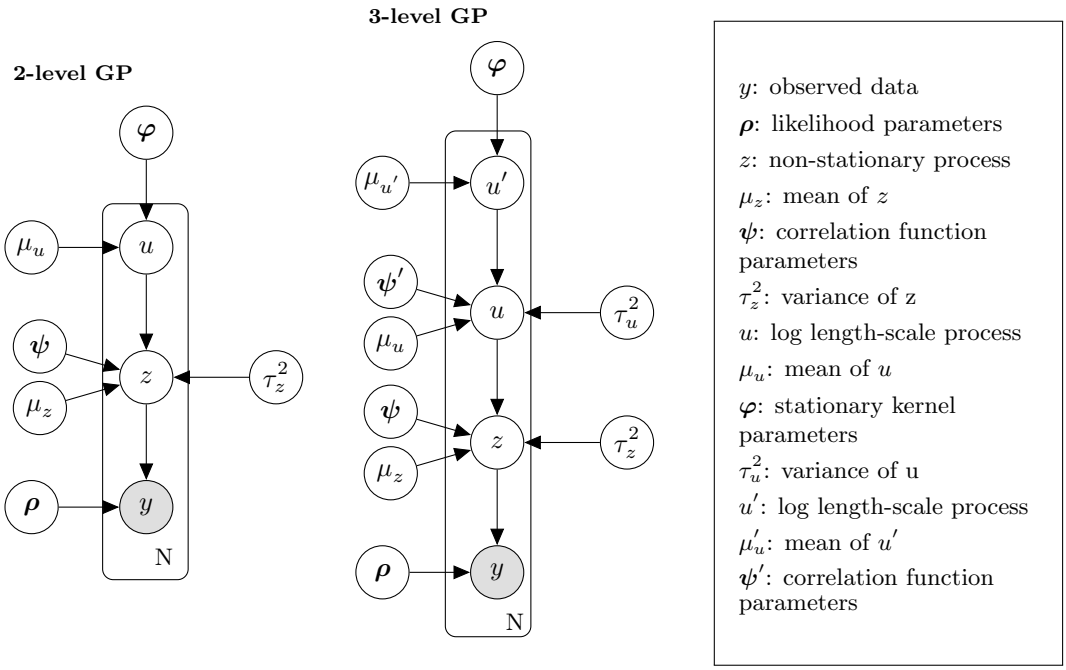


Figure 2.3: Plate diagram for non-stationary multi-level GP models.

The 2-level model described here naturally extends to multiple levels to construct the deep GP models discussed in Dunlop et al. (2018). This can be done by assigning a non-stationary prior for the log length-scale parameter. A graphical representation of this is given to the right of Figure 2.3, for the 3-level GP model.

2.3.1 Connection to deep Gaussian processes

It is worth mentioning that multi-level GPs are akin to the deep Gaussian processes (DGPs) introduced by Damianou and Lawrence (2013). Their approach employs function compositions over the inputs to create the hierarchy. In our case, the cascade of GPs is introduced to the model in the covariance function, as a prior distribution over spatially varying parameters, giving a full probabilistic nature to the model. Furthermore, compared to the DPGs, realisations of a multi-level GP model do not exhibit the pathologies described in Duvenaud et al. (2014). In fact, the realisations (see Figure 2.4) look comparable to what the authors termed input-connected networks, where each layer is dependent on the inputs and not only on the output of the previous layer.

Deep GPs have received increased interest in the literature in the past decade and proposals differ in how the layers are combined (e.g. Damianou and Lawrence,

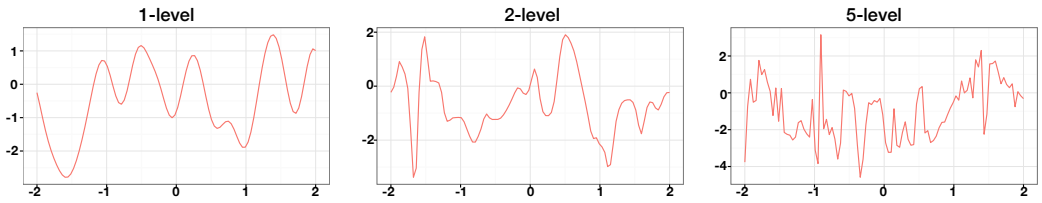


Figure 2.4: Draws from multi-level GP models employing a non-stationary squared exponential kernel with 1-level (stationary), 2-levels and 5-levels.

2013; Dunlop et al., 2018; Hegde et al., 2019; Blomqvist et al., 2018; Fox and Dunson, 2012). However, the key challenges, preventing wide-spread use of deep GPs, include developing interpretable constructions that lack degeneracy (Duvenaud et al., 2014), as well as developing efficient and scalable inference methods, despite the highly coupled layers and computational expense of GPs.

CHAPTER 3

CHALLENGES OF 2-LEVEL GP MODELS

While recognised as flexible constructions, multi-level Gaussian processes (GPs) raise interesting challenges on how to conduct inference efficiently from a Bayesian perspective. Such challenges include (i) inference over a high-dimensional posterior whose elements are strongly coupled, (ii) parameter identifiability, (iii) generalisation to D-dimensional settings, and (iv) high computational complexity. The discussion, initial comparisons, and findings here presented, serve as a stepping stone for the work introduced in the remaining chapters of the thesis.

3.1 Introduction

Consider the Gaussian process regression (GPR) model with Gaussian noise described in Section 2.1.1.1 and the non-stationary 2-level GP prior from Eq. (2.24) to define a 2-level GPR model for one-dimensional data through

$$\begin{aligned} y_n &\sim \mathrm{N}(y_n \mid z(x_n), \sigma_\varepsilon^2), \quad n = 1, ..., N, \\ z(\cdot) &\sim \mathrm{GP}\left(\mu_z, C_{\boldsymbol{\phi}}^{\mathrm{NS}}(\cdot,\cdot)\right), \\ u(\cdot) &\sim \mathrm{GP}\left(\mu_u, C_{\boldsymbol{\varphi}}^{\mathrm{S}}(\cdot,\cdot)\right), \\ (\sigma_\varepsilon^2, \boldsymbol{\psi}, \tau_z^2, \boldsymbol{\varphi}) &\sim \pi(\sigma_\varepsilon^2)\pi(\boldsymbol{\psi})\pi(\tau_z^2)\pi(\boldsymbol{\varphi}), \end{aligned} \tag{3.1}$$

with $\boldsymbol{\phi} = \{\tau_z^2, \boldsymbol{\psi}, u(\cdot)\}$ containing the parameters of the non-stationary kernel, $\boldsymbol{\varphi}$ denoting the parameters of the stationary covariance, μ_z and μ_u representing the prior mean for z and u, respectively. Because the prior over z is conjugate to the

likelihood, we can integrate out the latent function to obtain the target distribution:

$$\pi(\sigma_\varepsilon^2, \boldsymbol{u}, \boldsymbol{\psi}, \tau_z^2, \boldsymbol{\varphi} \mid X, \boldsymbol{y}) \propto \mathrm{N}(\boldsymbol{y} \mid \boldsymbol{\mu}_z, C_{\boldsymbol{\phi}}^{\mathrm{NS}} + \sigma_\varepsilon^2 I_N)\mathrm{N}(\boldsymbol{u} \mid \boldsymbol{\mu}_u, C_{\boldsymbol{\varphi}}^{\mathrm{S}})\pi(\sigma_\varepsilon^2)\pi(\boldsymbol{\psi})\pi(\tau_z^2)\pi(\boldsymbol{\varphi}), \tag{3.2}$$

where $\boldsymbol{\mu}_z$ and $\boldsymbol{\mu}_u$ denote N-dimensional vectors with elements all equal to μ_z and μ_u, respectively.

In standard, single-level GP models marginalisation of the latent function results in a posterior over a low-dimensional parameter space. While such parameters are usually estimated by maximising the marginal likelihood, also known as type II maximum likelihood (Titsias and Lázaro-Gredilla, 2013), Markov chain Monte Carlo (MCMC) inference is also attainable employing conventional schemes, such as Metropolis-Hastings (MH). In contrast, the marginal posterior for a 2-level GP involves a high-dimensional vector, $\boldsymbol{u}$, whose elements can be highly coupled and its dimension increases linearly with the number of data points. Moreover, there exists a strong correlation between the parameters $(\boldsymbol{u}, \boldsymbol{\psi}, \tau_z^2, \boldsymbol{\varphi})$, which can result in an MCMC chain with slow exploration and convergence to the target. Consequently, devising efficient MCMC samplers to explore the posterior in Eq. (3.2) can be extremely challenging.

Another critical obstacle for scalable Bayesian inference results from the fact that computing the posterior in Eq. (3.2) requires the evaluation of two N-dimensional Gaussian distributions involving matrix operations that scale cubically with the number of observations. Even in one-dimensional problems, the computational complexity constrains the applicability of the method to small or moderate size datasets. Further difficulties arise from extending 2-level GPs to D-dimensional settings. In full generality, the non-stationary kernel in Eq. (2.21) requires the estimation of N $(D \times D)$ matrices to describe the correlation structure of the field. This results in a highly parametrised, complicated model that hinders scalable inference.

We start this chapter by introducing a novel parametrisation of the kernel matrices for D-dimensional problems also presenting its corresponding hierarchical model. Section 3.3 presents an initial comparative evaluation on 1-D and 2-D synthetic datasets, which highlights the capabilities of the model but also the sampling difficulties. We continue in Section 3.4 with a discussion on hyperparameter identifiability, where we introduce an empirical-prior strategy to alleviate the problem. Section 3.5 covers a discussion on the computational complexity and emphasises the importance of efficient matrix algebra computations. We conclude this chapter with a summary of the main findings and remarks in Section 3.6.

3.2 Modelling the kernel matrices

As discussed in Chapter 2 (Section 2.3) the spatially varying kernel matrices are at the heart of the hierarchical structure of a 2-level GP. Existing approaches to extend the model beyond one-dimensional settings are based on spectral decompositions (Paciorek and Schervish, 2006; Neto et al., 2014; Risser and Calder, 2017), basis function representations (Katzfuss, 2013) or an isotropic assumption (Heinonen et al., 2016; Roininen et al., 2019). Here, we develop a general, flexible approach based on LDL^{T} factorisation (Karny, 2006) to model the spatially varying parameters. Without loss of generality, we focus on the two-dimensional setting.

3.2.1 LDL factorisation for the kernel matrices

In two-dimensional settings, employing an LDL^{T} parametrisation allows us to define a prior over the kernel matrices by utilising three GPs. First, we decompose the kernel matrix at location $\boldsymbol{x}_n$ as $\Sigma(\boldsymbol{x}_n) = L(\boldsymbol{x}_n)D(\boldsymbol{x}_n)L(\boldsymbol{x}_n)^T$, where

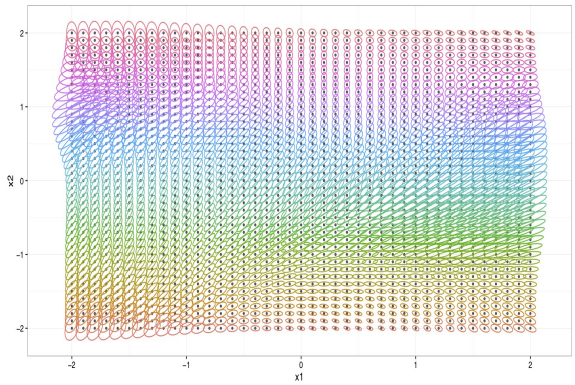
$$L(\boldsymbol{x}_n) = \begin{pmatrix} 1 & 0 \\ \ell_3(\boldsymbol{x}_n) & 1 \end{pmatrix}, \quad D(\boldsymbol{x}_n) = \begin{pmatrix} \ell_1^2(\boldsymbol{x}_n) & 0 \\ 0 & \ell_2^2(\boldsymbol{x}_n) \end{pmatrix},$$

with $\ell_1(\boldsymbol{x}_n) > 0, \ell_2(\boldsymbol{x}_n) > 0, \ell_3(\boldsymbol{x}_n) \in \mathbb{R}$, and $L(\boldsymbol{x}_n)^T$ denoting the transpose of $L(\boldsymbol{x}_n)$. This provides a unique factorisation of the kernel matrix, and in terms of the parameters $(\ell_1(\boldsymbol{x}_n), \ell_2(\boldsymbol{x}_n), \ell_3(\boldsymbol{x}_n))$, the kernel matrix at location $\boldsymbol{x}_n$ is given by,

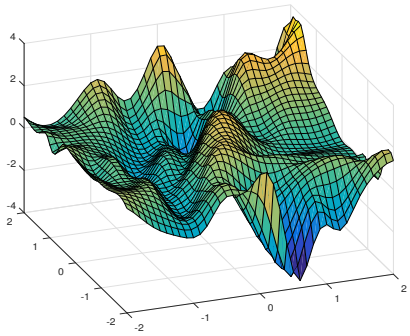
$$\Sigma(\boldsymbol{x}_n) = \begin{pmatrix} \ell_1^2(\boldsymbol{x}_n) & \ell_1^2(\boldsymbol{x}_n)\ell_3(\boldsymbol{x}_n) \\ \ell_1^2(\boldsymbol{x}_n)\ell_3(\boldsymbol{x}_n) & \ell_2^2(\boldsymbol{x}_n) + \ell_1^2(\boldsymbol{x}_n)\ell_3^2(\boldsymbol{x}_n) \end{pmatrix}. \tag{3.3}$$

In order to define a prior over the kernel matrices across the entire input space, we assume that $u_1 \coloneqq \log \ell_1(\cdot) \sim \mathrm{GP}\left(0, C^{\mathrm{S}}_{\boldsymbol{\varphi}_1}(\cdot,\cdot)\right)$, $u_2 \coloneqq \log \ell_2(\cdot) \sim \mathrm{GP}\left(0, C^{\mathrm{S}}_{\boldsymbol{\varphi}_2}(\cdot,\cdot)\right)$ and $\ell_3(\cdot) \sim \mathrm{GP}\left(0, C^{\mathrm{S}}_{\boldsymbol{\varphi}_3}(\cdot,\cdot)\right)$. More precisely, the hierarchical structure of the 2-level GPR model in 2-dimensions is

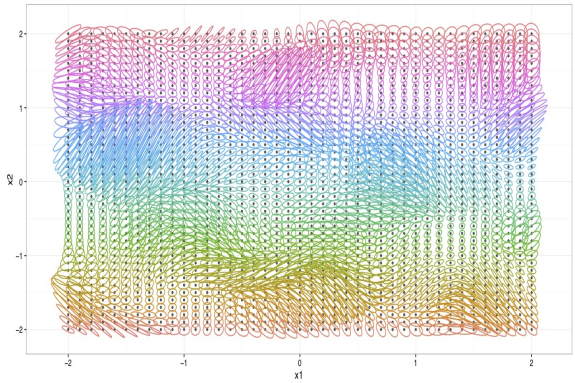
$$\begin{aligned} y_n &\sim \mathrm{N}(y_n \mid z(\boldsymbol{x}_n), \sigma_\varepsilon^2) \quad n = 1, \ldots, N \\ z(\cdot) &\sim \mathrm{GP}\left(\mu_z, C^{\mathrm{NS}}_{\boldsymbol{\phi}}(\cdot,\cdot)\right) \\ u_1(\cdot) := \log \ell_1(\cdot) &\sim \mathrm{GP}\left(0, C^{\mathrm{S}}_{\boldsymbol{\varphi}_1}(\cdot,\cdot)\right), \\ u_2(\cdot) := \log \ell_2(\cdot) &\sim \mathrm{GP}\left(0, C^{\mathrm{S}}_{\boldsymbol{\varphi}_2}(\cdot,\cdot)\right), \\ \ell_3(\cdot) &\sim \mathrm{GP}\left(0, C^{\mathrm{S}}_{\boldsymbol{\varphi}_3}(\cdot,\cdot)\right), \\ (\sigma_\varepsilon^2, \boldsymbol{\psi}, \tau_z^2, \boldsymbol{\varphi}_1, \boldsymbol{\varphi}_2, \boldsymbol{\varphi}_3) &\sim \pi(\sigma_\varepsilon^2)\pi(\boldsymbol{\psi})\pi(\tau_z^2)\pi(\boldsymbol{\varphi}_1)\pi(\boldsymbol{\varphi}_2)\pi(\boldsymbol{\varphi}_3), \end{aligned} \tag{3.4}$$



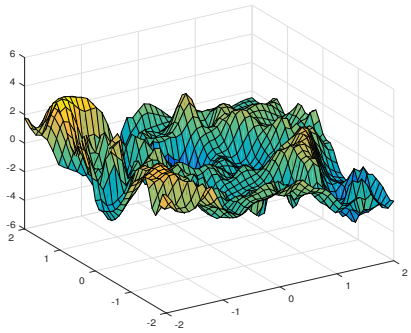
(a) Kernel matrices



(b) Non-stationary realisation



(c) Kernel matrices



(d) Non-stationary realisation

Figure 3.1: Effect of the kernel matrices on realisations of the non-stationary process. (Top row:) To the right, the latent field of kernel matrices, where second level stationary GPs are Matérn with $\lambda_1 = 1.1$, $\lambda_2 = 2$, $\lambda_3 = 1.5$, $\nu_{u_j} = 30$ and $\tau^2_{u_j} = 1$ for $j = 1, 2, 3$. To the left, a realisation from a non-stationary Matérn covariance with kernel matrices (a), $\tau^2_z = 1.5^2$ and $\nu_z = 15$. (Bottom row:) To the right, latent field of kernel matrices, where stationary Matérn parameters are set to $\lambda_j = 0.5$, $\nu_{u_j} = 30$ and $\tau^2_{u_j} = 1$ for $j = 1, 2, 3$. To the left, a realisation from a non-stationary Matérn covariance with kernel matrices (c), $\tau^2_z = 1.5^2$ and $\nu_z = 15$.

where $\boldsymbol{\phi} = \{\tau^2_z, \boldsymbol{\psi}, \Sigma(\cdot)\}$. In this setting, each component defining the kernel matrices has a different GP prior with its own set of hyperparameters. Notice that this formulation contains several special cases. For example, setting $\ell_3(\cdot)$ equal zero imposes a diagonal kernel matrix with independent, dimension-specific GP priors on the log length-scale processes. Instead, a non-stationary isotropic kernel is recovered when $\ell_2(\cdot)$ equals zero and $\ell_3(\cdot)$ equals one. To gain some intuition on how the kernel matrices affect realisations of the process, we illustrate in Figure 3.1 two latent fields of kernel matrices and realisations from a Gaussian process with non-stationary

Matérn covariance that employs such kernel matrices.

3.3 Empirical evaluation

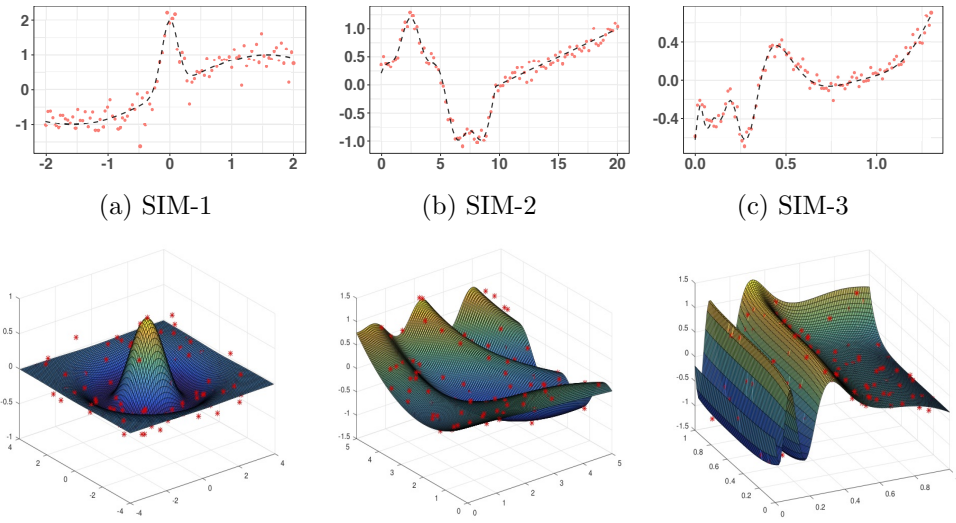
We present an empirical evaluation on 1-D and 2-D synthetic datasets, comparing posterior inference and predictions from three different GP models. The first model corresponds to a single-level, stationary GP regression model with a Matérn covariance function (STAT). The second is a single-level, non-stationary GP regression model based on the neural network kernel (NN). The third is a 2-level, non-stationary model employing the non-stationary Matérn kernel. Notice that we select a neural network covariance function as a benchmark because it requires few parameters, leading to a non-stationary model at no extra computational cost over a stationary model.

The objective of this study is twofold: (i) devising an effective sampler for the spatially varying parameter, and (ii) obtaining initial comparisons of the predictive performance of the 2-level GP model. To focus on (i), we fix the hyperparameters of the second level GP in both 1-D and 2-D settings, simplifying the hierarchical constructions shown in Eq.(3.1) and Eq. (3.4).

The one-dimensional data-generating functions employed are: (i) a smooth function taken from Paciorek (2003) (SIM-1), (ii) a discontinuous function from Gramacy (2007) (SIM-2), and (iii) a non uniformly smooth function from Xiong et al. (2007) (SIM-3). Figure 3.2(a)-(c) illustrates the generated data, where we use $N = 100$ for each dataset. For the 2-D examples, the first dataset (SIM-4) corresponds to $N = 120$ noisy points employing the Ricker wavelet as the latent function. The second (SIM-5) represents $N = 130$ points utilising the latent function from Xiong et al. (2007). The third (SIM-6) consists of $N = 130$ noisy observations generated with a modification of a function employed by Xiong et al. (2007). Moreover, for these three datasets, the coordinates in the training and test sets are selected using Latin hypercube sampling.[†] The generated datasets and the corresponding true functions are shown in Figure 3.2(d)-(f) (see Appendix B.1 for the explicit formulation of the functions employed).

To compare the predictive performance of STAT, NN and the 2-level GP, we perform out-of-sample predictions at locations $X^* = (\boldsymbol{x}_1^*, \ldots, \boldsymbol{x}_{N^*}^*)$. To evaluate the

[†] For the Latin hypercube sampling we employ the R-package by Carnell (2016).



(a) SIM-1 (b) SIM-2 (c) SIM-3

(d) SIM-4 (e) SIM-5 (f) SIM-6

Figure 3.2: Synthetic datasets. (a)-(c) One-dimensional data, noisy observations in red and true function shown with a dashed line. (d)-(f): Two-dimensional datasets where red crosses denote the noisy observations and the curve depicts the true function.

point prediction accuracy, we employ the root mean squared error,

$$\text{RMSE} = \sqrt{\frac{1}{N^*}\sum_{i=1}^{N^*}(z(\boldsymbol{x}_i^*) - \bar{z}(\boldsymbol{x}_i^*))^2},$$

and the mean absolute error,

$$\text{MAE} = \frac{1}{N^*}\sum_{i=1}^{N^*}|z(\boldsymbol{x}_i^*) - \bar{z}(\boldsymbol{x}_i^*)|,$$

where $z(\boldsymbol{x}_i)$ denotes the true value of the latent function at location $\boldsymbol{x}_i$ and $\bar{z}(\boldsymbol{x}_i)$ the obtained prediction at such location. In addition, to account for uncertainty in the predictive performance, we include the negative log pointwise predictive density (NLPPD) of the true targets at new locations,

$$\text{NLPPD} = -\sum_{i=1}^{N^*}\log\left(\frac{1}{T}\sum_{t=1}^{T}\pi({y_i}^*|X, \boldsymbol{y}, x_i^*, \sigma_\varepsilon^{2(t)}, \boldsymbol{\phi}^{(t)})\right),$$

where $\sigma_\varepsilon^{2(t)}$ and $\boldsymbol{\phi}^{(t)}$ denote the t^{th} sample from the marginal posterior $\pi(\sigma_\varepsilon^2, \boldsymbol{\phi} \mid$

$X, \boldsymbol{y}$), and $\pi(y_i^*|X, \boldsymbol{y}, x_i^*, \sigma_\varepsilon^{2(t)}, \boldsymbol{\phi}^{(t)})$ is the conditional predictive distribution (see Section 2.1.1.1).

We start by discussing the selected priors and hyperpriors and the inference procedure for each of the models in Section 3.3.1, and in Section 3.3.2, we present the results for out-of-sample predictions in all six simulated datasets.

3.3.1 Prior specification and posterior inference

In the following, we provide further details on prior and hyperprior specification and describe the inference scheme employed for each of the three models analysed: STAT, NN and 2-level GP. For all models and experiments, we select non-informative priors for the noise and magnitude parameters, defined on the log space, namely $\log \sigma_\varepsilon^2 \sim \mathrm{N}(0, 3^2)$, $\log \tau^2 \sim \mathrm{N}(0, 3^2)$. Traceplots are shown in Appendix B.2.

STAT: For all one-dimensional datasets, the prior for the latent function is a zero-centred GP with the stationary Matérn covariance function from Eq.(2.17), which has hyperparameters (ν, τ^2, λ). In simulation studies, we found that the smoothness parameter is difficult to recover; therefore, we set it to $\nu = 15$ for SIM-1, and $\nu = 3/2$ for both SIM-2, and SIM-3. We select a non-informative prior for the length-scale, with $\lambda \sim \mathrm{Gam}(1, 0.09)$. In two-dimensional settings, we employ the anisotropic version of the Matérn covariance function, with different length-scales for each dimension. The smoothness parameter is fixed at $\nu = 15$ for both SIM-4 and SIM-5 and $\nu = 3/2$ for SIM-6. The prior on each length-scale, λ_1 and λ_2 is defined as in the one-dimensional setting.

After marginalisation of the latent function, the target distribution is

$$\pi(\sigma_\varepsilon^2, \tau^2, \boldsymbol{\lambda} \mid \boldsymbol{y}, X) \propto \pi(\boldsymbol{y} \mid X, \sigma_\varepsilon^2, \tau^2, \boldsymbol{\lambda}) \pi(\log \sigma_\varepsilon^2) \pi(\log \tau^2) \pi(\boldsymbol{\lambda}),$$

where $\boldsymbol{\lambda} := \lambda$ in one-dimensional settings, and $\boldsymbol{\lambda} := (\lambda_1, \lambda_2)$ for two-dimensional datasets. To sample from the target distribution, we employ an Metropolis-within-Gibbs (MWG) algorithm. We employ random-walk proposals for all parameters, with scale parameters tuned manually for each dataset in order to achieve an optimal acceptance rate. The algorithm is iterated for $T = 250,000$, with a burn-in period of $5,000$ to reduce the effect of the starting values and a thinning factor of 5 to reduce the autocorrelation.

NN: The prior distribution for the latent function is a zero-centred GP with the neural network kernel shown in Eq.(2.20), which has hyperparameters $(\tau^2, \sigma_0^2, \sigma_1^2)$

in 1-D and $(\tau^2, \sigma_0^2, \sigma_1^2, \sigma_2^2)$ in 2-D. For σ_0^2, σ_1^2 and σ_2^2, we define diffuse but proper priors in the log space; namely, $\mathrm{N}(0, 9^2)$, to reflect that in our experience the values of the weight and bias variances may be substantially large. After integration with respect to the latent function, the target distribution for the neural network GP model is

$$\pi(\sigma_\varepsilon^2, \tau^2, \sigma_0^2, \sigma_1^2, \sigma_2^2 \mid \boldsymbol{y}, X) \propto \pi(\boldsymbol{y} \mid X, \sigma_\varepsilon^2, \tau^2, \sigma_0^2, \sigma_1^2, \sigma_2^2)\pi(\log \sigma_\varepsilon^2)\pi(\log \tau^2)\pi(\log \sigma_0^2) \\ \pi(\log \sigma_1^2)\pi(\log \sigma_2^2).$$

For inference, we employ an MWG algorithm. During the first k iterations, we use random-walk proposals for each of the parameters, with the scale tuned manually to get an optimal acceptance rate. Due to the high correlation between $(\sigma_0^2, \sigma_1^2, \sigma_2^2)$, we employ a multivariate random-walk proposal for the remaining $T - k$ iterations with an empirical covariance matrix calculated from the first k iterations. The algorithm is iterated for $T = 240,000$, with a burnin of $80,000$ and a thinning factor of 4.

2-level GP: The hierarchical construction of this model is defined in Eq. (3.1) for 1-D and in Eq. (3.4) for 2-D settings. The prior of the latent function is a zero-mean GP with the non-stationary Matérn covariance function from Eq. (2.22), which has parameters τ_z^2, ν_z, $\Sigma(\boldsymbol{x}_n)$, for $n = 1, \dots N$. Furthermore, the smoothness parameter ν_z is fixed for each dataset, as in the stationary model.

For one-dimensional datasets, the prior for the spatially varying parameter is a zero-centred GP with a Matérn stationary kernel with hyperparameters $\boldsymbol{\varphi} = (\nu_u, \tau_u^2, \lambda)$. We fix $\nu_u = 30$ to ensure the elements of $\boldsymbol{u}$ to vary smoothly, while the length-scale λ_u and magnitude τ_u^2 are fixed according to a grid search over $[.5, 3.5] \times [.5, 3.5]$. For SIM-1, $\lambda = 0.5$ and $\tau_u^2 = 2$; for SIM-2, $\lambda = 2.5$ and $\tau_u^2 = 1$; and for SIM-3, $\lambda = 0.5$ and $\tau_u^2 = 2.5$. The target distribution is

$$\pi(\sigma_\varepsilon^2, \tau_z^2, \boldsymbol{u} \mid \boldsymbol{y}, X) \propto \pi(\boldsymbol{y} \mid X, \sigma_\varepsilon^2, \tau_z^2, \boldsymbol{u})\pi(\log \sigma_\varepsilon^2)\pi(\log \tau_z^2)\pi(\boldsymbol{u}). \tag{3.5}$$

Sampling from Eq. (3.5) is challenging. Simulation studies were implemented to assess the performance of different sampling methodologies. The general approach is a MWG scheme, which employs MH steps for each of the components in the target distribution. In each algorithm, adaptive random-walk steps for $\log \sigma_\varepsilon^2$ and $\log \tau_z^2$ provided good mixing. However, devising an efficient scheme for the multi-dimensional vector $\boldsymbol{u}$ is proved difficult because its elements, by construction, are highly correlated. We experimented with different methods to sample $\boldsymbol{u}$. First, we investigated the performance of three MH schemes: (i) the adaptive Metropolis

(AM) in Roberts and Rosenthal (2009), which employs a mixture of Gaussians as proposal distribution, (ii) an MWG scheme with adaptive random walks for each element of $\boldsymbol{u}$ and (iii) an MH step that employs the prior as proposal distribution. However, these schemes proved inefficient, either resulting in a random-walk behaviour or demanding long runs to reach convergence. Second, we implemented a no-u-turn Hamiltonian Monte Carlo (HMC) sampler (Hoffman and Gelman, 2014). HMC is robust to high-dimensions and high correlations, providing good mixing of the chains; nonetheless, it requires the derivative of the log marginal likelihood with respect to $\boldsymbol{u}$, which in turn needs the derivative of the non-stationary covariance function employed. For the Matérn kernel, this implies computing the derivative of a Bessel function, which results in two Bessel function evaluations, a slow and numerically unstable operation (Harrison, 2009). Nevertheless, an HMC sampler may be suitable for other kernels, such as the non-stationary version of the squared exponential (see Heinonen et al. (2016)), whose derivative has a simplier form. Moreover, one could employ automatic differentiation tools to avoid analytical derivatives. Finally, the selected scheme employs elliptical slice sampling (Ell-SS) (Murray et al., 2010). This improves the mixing, increases the speed of convergence and reduces the autocorrelation of the chains, while avoiding derivative computations and parameter tuning.

In two-dimensional problems, the (2×2) kernel matrices are parametrised through $\boldsymbol{u}_1, \boldsymbol{u}_2$ and $\boldsymbol{\ell}_3$, which are assigned independent, zero mean GPs priors with a stationary Matérn kernel. As for 1-D datasets, the hyperparameters of the stationary kernel are fixed. For SIM-4, we fix $\nu_{u_j} = 30$, $\tau^2_{u_j} = 3.5$ and $\lambda_j = 4.5$ for $j = 1, 2, 3$. For SIM-5, we set $\nu_{u_j} = 30$, $\tau^2_{u_j} = 1$ and $\lambda_j = 2.5$ for $j = 1, 3$; whereas $\nu_{u_2} = 30$, $\tau^2_{u_2} = 1$ and $\lambda_2 = 1.5$. For SIM-6, $\nu_{u_j} = 30$, $\tau^2_{u_j} = 1$ and $\lambda_j = 0.5$ for $j = 1, 3$; whereas $\nu_{u_2} = 30$, $\tau^2_{u_2} = 1$ and $\lambda_2 = 1.5$. The posterior of interest for two-dimensional problems is

$$\pi(\sigma^2_\varepsilon, \tau^2_z, \boldsymbol{u}_1, \boldsymbol{u}_2, \boldsymbol{\ell}_3 \mid \boldsymbol{y}, X) \propto \pi(\boldsymbol{y} \mid X, \sigma^2_\varepsilon, \tau^2_z, \boldsymbol{u}_1, \boldsymbol{u}_2, \boldsymbol{\ell}_3)\pi(\log \sigma^2_\varepsilon)\pi(\log \tau^2_z)\pi(\boldsymbol{u}_1) \\ \pi(\boldsymbol{u}_2)\pi(\boldsymbol{\ell}_3).$$

In this setting, we jointly update $\boldsymbol{u}_1, \boldsymbol{u}_2$ and $\boldsymbol{\ell}_3$ using Ell-SS. For one-dimensional simulated experiments, the algorithm was iterated for $T = 65,000$, with a burn-in of $5,000$ and a thinning factor of 2. For two-dimensional datasets, the algorithm was iterated for $T = 20,000$, with a burn-in of $3,000$. This difference is due to the computational cost of the model.

3.3.2 Predictions

For STAT and NN, we employ Eq. (2.15) to obtain the predictive mean of the latent function. Predictions for the 2-level GP model require integration of the spatially varying parameters at the new locations with respect to their predictive distribution. For instance, in one-dimensional problems, the predictive distribution of $\boldsymbol{z}^*$ can be computed through

$$\pi(\boldsymbol{z}^* \mid X, \boldsymbol{y}, X^*) \approx \frac{1}{T}\sum_{t=1}^{T}\frac{1}{J}\sum_{j=1}^{J}\pi(\boldsymbol{z}^* \mid X, \boldsymbol{y}, X^*, \sigma_\varepsilon^{2(t)}, \tau_z^{2(t)}, \boldsymbol{u}^{*(j,t)}), \tag{3.6}$$

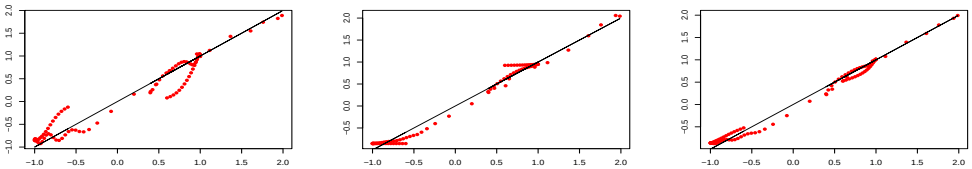
where $\sigma_\varepsilon^{2(t)}, \tau_z^{2(t)}$ represent the t^{th} MCMC sample and $\boldsymbol{u}^{*(j,t)}$ is the j^{th} draw from $\pi(\boldsymbol{u}^* \mid \boldsymbol{u}^{(t)}, \boldsymbol{\varphi})$. Predictions in two-dimensional settings are analogous but require integration of $\boldsymbol{u}_1^*$, $\boldsymbol{u}_2^*$ and $\boldsymbol{\ell}_3^*$.

For the three one-dimensional datasets, we perform out-of-sample predictions at $N^* = 100$ locations, whereas for 2-D settings, we use $N^* = 90$ locations. The results illustrate that STAT tends to over-fit where the true function is smooth, due to a small inferred length scale in order to capture sharp changes in other regions. Secondly, the neural network model shows a reasonable performance for the three one-dimensional datasets, representing an improvement over the stationary model. However, the 2-level GP tends to perform better at recovering peaks (see SIM-1 SIM-2 in Figure B.6 in the Appendix). For SIM-5, STAT and NN perform very similar; whereas the 2-level GP model does a better job (see the region $[(0,1) \times (0,2)]$ in Figure B.7 in the Appendix). For SIM-6, both non-stationary models achieve a better fit to the true surface. For ease of visualisation and comparison, we show in Figure 3.3 scatter plots of predicted versus true values.

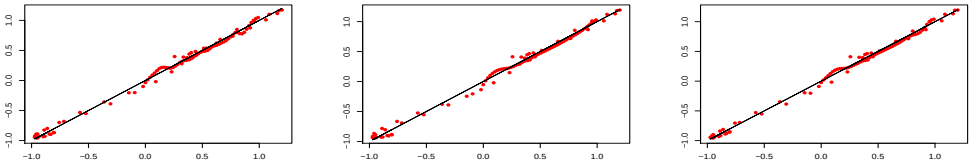
Additionally, we display in Figure 3.4, the estimated kernel matrices in the 2-level GP model to illustrate how their evolution in the input space resembles the covariance structure of the true function. Table 3.1 summarises the predictive performance in terms of RMSE, MAE, and NLPPD. We observe that in terms of RMSE and MAE, the best performance is attained with the 2-level GP model in all six datasets, whereas the NLPPD slightly favours the NN model for SIM-2.

3.4 Inferring the hyperparameters

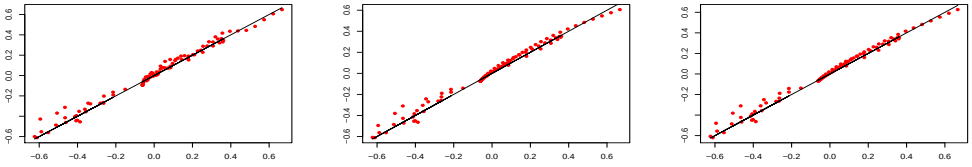
In our simulation studies from Section 3.3, we found that the model is sensitive to the choice of hyperparameters in the second level GP. Although this flaw was



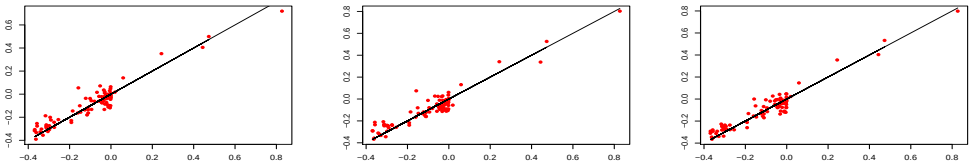
(a) SIM-1 with STAT (b) SIM-1 with NN (c) SIM-1 with 2-level



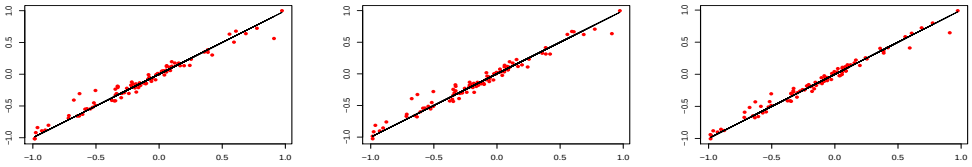
(d) SIM-2 with STAT (e) SIM-2 with NN (f) SIM-2 with 2-level



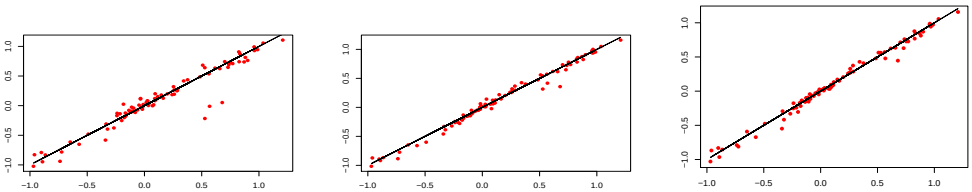
(g) SIM-3 with STAT (h) SIM-3 with NN (i) SIM-3 with 2-level



(j) SIM-4 with STAT (k) SIM-4 with NN (l) SIM-4 with 2-level

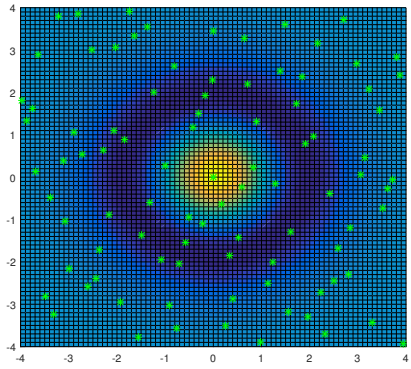


(m) SIM-5 with STAT (n) SIM-5 with NN (o) SIM-5 with 2-level

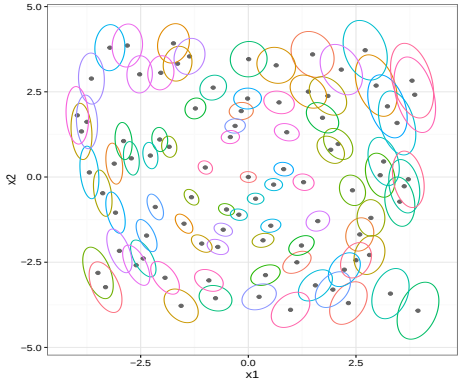


(p) SIM-6 with STAT (q) SIM-6 with NN (r) SIM-6 with 2-level

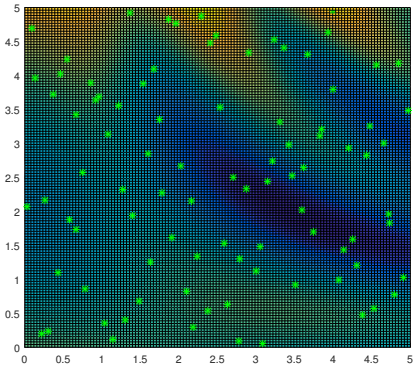
Figure 3.3: Scatter plots of true versus predicted values for the six synthetic datasets. Each row represents one of the simulated examples and the columns corresponds to the three different models.



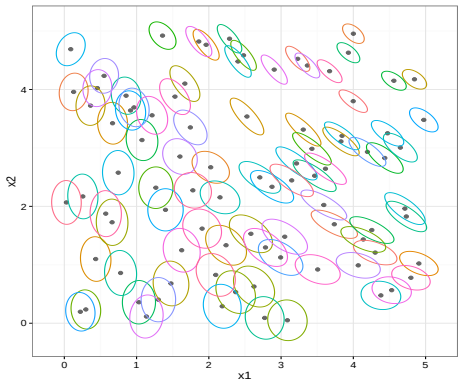
(a) True function for SIM-4



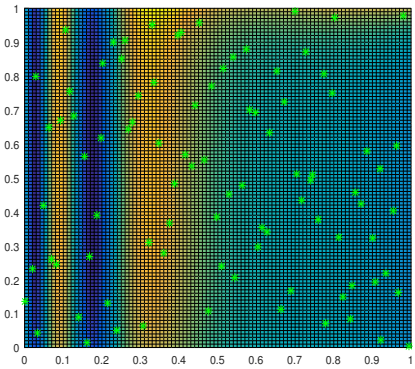
(b) Predicted kernel matrices for SIM-4



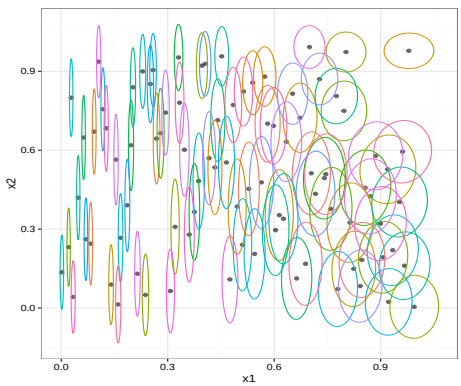
(c) True function for SIM-5



(d) Predicted kernel matrices for SIM-5



(e) True function for SIM-6



(f) Predicted kernel matrices for SIM-6

Figure 3.4: Estimated kernel matrices at test locations for 2-D datasets. The first column denotes the true function for each of the simulated datasets with test locations depicted as green stars. The second column illustrates the estimate of the kernel matrices at the test locations (arbitrary colors).

	RMSE			**MAE**			**NLPPD**		
	STAT	NN	2-level	STAT	NN	2-level	STAT	NN	2-level
SIM-1	0.2096	0.1110	**0.0943**	0.1523	0.0879	**0.0777**	39.17	30.16	**29.22**
SIM-2	0.0469	0.0498	**0.0440**	0.0392	0.0408	**0.0374**	-79.75	**-80.66**	-80.20
SIM-3	0.0406	0.0376	**0.0360**	0.0284	0.0270	**0.0251**	-117.62	-119.58	**-121.38**
SIM-4	0.0568	0.0607	**0.0530**	0.0429	0.0471	**0.0418**	- 81.76	-79.89	**-82.87**
SIM-5	0.0828	0.0816	**0.0649**	0.0553	0.0579	**0.0467**	-65.64	-63.66	**-77.22**
SIM-6	0.1405	0.0627	**0.0542**	0.0767	0.0404	**0.0379**	-37.36	-64.78	**-69.75**

Table 3.1: Predictive performance for the six simulated datasets under the three different models (STAT, NN and 2-level GP). Best value in boldface.

already reported (Neto et al., 2014), fixing hyperparameters when employing the 2-level GP model is common (see Heinonen et al., 2016; Paciorek and Schervish, 2006; Roininen et al., 2019). Additionally, even though our initial study in Section 3.3 results in reasonable predictions, inferring the hyperparameters can further improve predictive performance. Moreover, this is a necessary step to extend the hierarchy to deeper constructions.

Consider again the posterior distribution from the one-dimensional hierarchical model given in Eq. (3.2). Two challenges prevent efficient inference of the hyperparameters. First, strong dependency between the log length-scale process $\boldsymbol{u}$ and the hyperparameters $\boldsymbol{\varphi}$ can result in chains with poor mixing that converge rather slowly to the stationary distribution. Second, covariance hyperparameters present identifiability issues (Zhang, 2004).

To tackle the first issue, we use the non-centred parametrisation of Papaspiliopoulos et al. (2007). Thus, we re-write the posterior of interest in terms of new random variables $\boldsymbol{\zeta} \sim \mathrm{N}(0, I_N)$ and define $\boldsymbol{u} = \mathrm{chol}(C^{\mathrm{S}}_{\boldsymbol{\varphi}})\boldsymbol{\zeta} + \boldsymbol{\mu}_u$. The re-parametrised posterior has the form

$$\pi(\sigma^2_\varepsilon, \boldsymbol{\zeta}, \boldsymbol{\psi}, \tau^2_z, \boldsymbol{\varphi} \mid X, \boldsymbol{y}) \propto \mathrm{N}(\boldsymbol{y} \mid \boldsymbol{\mu}_z, C^{\mathrm{NS}}_{\boldsymbol{\phi}'} + \sigma^2_\varepsilon I_N)\mathrm{N}(\boldsymbol{\zeta} \mid 0, I_N)\pi(\sigma^2_\varepsilon)\pi(\boldsymbol{\psi})\pi(\tau^2_z)\pi(\boldsymbol{\varphi}),$$

with $\boldsymbol{\phi}' = \{\tau^2_z, \psi, \boldsymbol{\zeta}\}$.

Regarding the second issue, there are two sources of hyperparameter non identifiability in the model. The first one arises as a result of the interaction between the magnitude and length-scale parameters in the covariance function. Zhang (2004) pointed out that for the stationary Matérn family, τ^2_u and λ are not consistently estimable under fixed domain asymptotics; however, for a fixed ν the quantity $\tau^2_u/\lambda^{2\nu}$ is identifiable. Therefore, one can employ re-parametrisations of the hyperparameters during the inference procedure (e.g. Christensen et al., 2006, Section 3.2). In

general, which parameters are consistently estimable and which re-parametrisation can be used, depend on the kernel employed and there are currently no general guidelines (Fuglstad et al., 2019). The second source of non-identifiability results from employing non-informative priors for the hyperparameters. Because the observed data do not provide information about the hyperparameter values beyond the observed domain and range, the posterior in that case is only informed through the prior. Thus, a naive implementation that assigns extremely broad priors can result in unreasonable inferences. For instance, one can obtain posterior estimates of λ that are greater than the data domain. Moreover, the MCMC chain can get trapped and spend too much time exploring values outside of the data domain, due to equivalent likelihood evaluations in this range. To address this, we propose to use the observed data to fix the second level magnitude τ_u^2 and to constrain the prior information of $\boldsymbol{z}$, τ_z^2, $\boldsymbol{u}$ and λ.

3.4.1 Empirical priors

We provide guidelines to fix the mean of $\boldsymbol{z}$, the mean and variance of $\boldsymbol{u}$, and we discuss prior distributions for $\log \tau_z^2$ and $\log \lambda$. Note that here we work with priors in the logarithm scale to employ adaptive random walk Metropolis-Hastings (RW-MH) steps. However, a similar approach can be utilised with other priors and proposal mechanisms.

First, for the non-stationary process, one can re-scale the data to have zero mean and unit variance; such that $\boldsymbol{z} \sim \mathrm{N}(0, C_{\boldsymbol{u}}^{\mathrm{NS}})$ and $\tau_z = 1$ is an appropriate assumption, where we use $C_{\boldsymbol{u}}^{\mathrm{NS}}$ to highlight that $\boldsymbol{u}$ is the only parameter to be estimated in the non-stationary covariance. Otherwise, if one aims to use the prior $\boldsymbol{z} \sim \mathrm{N}(\boldsymbol{\mu}_z, C_{\boldsymbol{\phi}}^{\mathrm{NS}})$, the observed response can be used to fix $\boldsymbol{\mu}_z$ and to determine the parameters of a Gaussian prior for $\log \tau_z^2 \sim \mathrm{N}(\mu_{\tau_z}, \sigma_{\tau_z}^2)$. The elements of $\boldsymbol{\mu}_z$ can be set equal to the mean of $\boldsymbol{y}$. For the magnitude parameter, we aim to ensure that most of the prior mass lies within the range of the response variable. Accordingly, we define γ to be the range of $\boldsymbol{y}$ and propose to define the parameters of the Gaussian prior by assuming $\Pr(l_z \leq \tau_z^2 \leq \gamma^2/4) = 0.95$, where l_z represents a suitable lower bound on the magnitude. Using the quantile function of a Gaussian random variable, we obtain the system of equations:

$$\begin{aligned} \mu_{\tau_z} - 1.96\sigma_{\tau_z} &= \log l_z, \\ \mu_{\tau_z} + 1.96\sigma_{\tau_z} &= 2\log\left(\frac{\gamma}{4}\right), \end{aligned}$$

which can be solved to determine μ_{τ_z} and σ_{τ_z}.

Second, for the spatially varying log length-scale prior, $\boldsymbol{u} \sim \mathcal{N}(\boldsymbol{\mu}_u, C^{\mathrm{S}}_{\boldsymbol{\varphi}})$, we propose to empirically fix its mean and magnitude, thus only inferring the length-scale λ. We start by computing the minimum and maximum covariate distance, given respectively by

$$\alpha = \min_{\boldsymbol{x}_i, \boldsymbol{x}_j \in X, \boldsymbol{x}_i \neq \boldsymbol{x}_j} \|\boldsymbol{x}_i - \boldsymbol{x}_j\| \quad \text{and} \quad \beta = \max_{\boldsymbol{x}_i, \boldsymbol{x}_j \in X, \boldsymbol{x}_i \neq \boldsymbol{x}_j} \|\boldsymbol{x}_i - \boldsymbol{x}_j\|,$$

where for notational simplicity, we focus on an isotropic kernel with a single length-scale process. Because identifiability issues arise for length scales outside of $[\alpha, \beta]$, we aim to place most of the prior mass within this range for each ℓ_n, by assuming $\Pr(\alpha \leq \ell_n \leq \beta) = 0.95$. To accomplish this, we solve:

$$\begin{aligned} \mu_u - 1.96\tau_u &= \log \alpha, \\ \mu_u + 1.96\tau_u &= \log \beta, \end{aligned}$$

to find μ_u and τ_u^2.

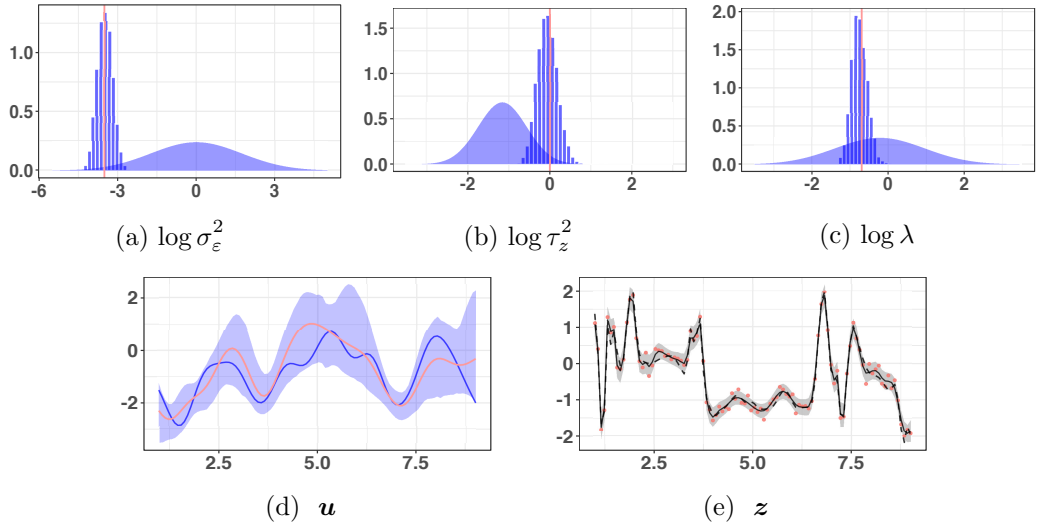
Finally, the same approach can be used to set the parameters of Gaussian prior for $\log \lambda \sim \mathrm{N}(\mu_\lambda, \sigma_\lambda^2)$, by solving:

$$\begin{aligned} \mu_\lambda - 1.96\sigma_\lambda &= \log \alpha, \\ \mu_\lambda + 1.96\sigma_\lambda &= \log \beta. \end{aligned}$$

3.4.2 Parameter recovery

We study parameter recovery in the 2-level GP model, employing a non-centred parametrisation for $\boldsymbol{u}$ and the empirical priors discussed above. To this aim, we simulate $N = 100$ observations in the domain $[1, 9]$ using the model in Eq. (3.1) with a squared exponential (SE) kernel for both stationary and non-stationary processes. The true parameters are set to $\sigma_\varepsilon^2 = 0.03$, $\tau_z^2 = 1$, $\tau_u^2 = 2$ and $\lambda = 0.5$. The empirical approach results in the following priors: $\boldsymbol{z} \sim \mathrm{N}(0, C^{\mathrm{NS}}_{\boldsymbol{\phi}})$, $\log \tau_z^2 \sim \mathrm{N}(-1.152, 0.344)$, $\boldsymbol{u} \sim \mathrm{N}(-0.218, C^{\mathrm{S}}_{\lambda})$, $\tau_u^2 = 1.374$, and $\log \lambda \sim \mathrm{N}(-0.218, 1.374)$. We use a MWG scheme to iterate over the components of the posterior $\pi(\sigma_\varepsilon^2, \tau_z^2, \boldsymbol{\zeta}, \lambda \mid X, \boldsymbol{y})$, employing adaptive RW-MH steps for $\sigma_\varepsilon^2, \tau_z^2$, and λ and an Ell-SS step for $\boldsymbol{\zeta}$. Figures 3.5(a)-(c) illustrate that the the posterior concentrates around the true values of $\log \sigma_\varepsilon^2$, $\log \tau_z^2$, and $\log \lambda$. Similarly, Figure 3.5(d) depicts the recovery of the spatially varying parameter by illustrating its posterior mean with 95% credible intervals versus the truth. Additionally, Figure 3.5(e) shows how the posterior mean

of $\boldsymbol{z}$ recovers the latent function of interest.



(a) $\log \sigma_\varepsilon^2$ (b) $\log \tau_z^2$ (c) $\log \lambda$

(d) $\boldsymbol{u}$ (e) $\boldsymbol{z}$

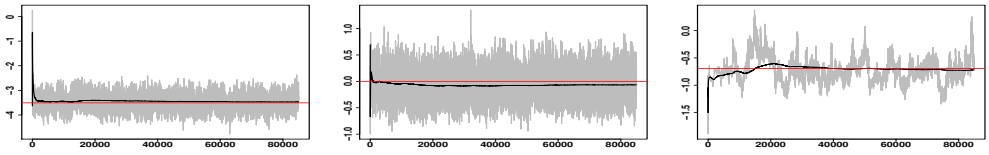
Figure 3.5: Recovery of the spatially varying parameter and the non-stationary function. (a)-(c) Prior and posterior densities for $\log \sigma_\varepsilon^2$, $\log \tau_z^2$ and $\log \lambda$. Histograms depict samples from the posterior distribution, density plots shows the empirical prior employed, and the red line corresponds to the true parameter value. (d) Recovery of the spatially varying parameter. The red line depicts the true parameter and the purple line its estimated posterior mean with 95% credible intervals. (e) Recovery of the non-stationary function. True function shown with a dashed line with observed data in red versus the posterior estimate of $\boldsymbol{z}$ with 95% credible intervals in grey.

To illustrate the performance of the sampler, we show convergence of the chains (Figure 3.6) by depicting traceplots and plots of the cumulative averages for some of the parameters in the model.

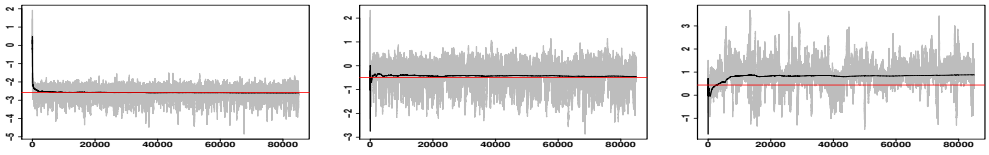
To highlight the dangers of arbitrarily fixing the hyperparameters, we repeat the experiment described above with the second level magnitude set at $\log \tau_u^2 = 0.1$ and the mean of the log length-scale process at $\mu_u = 0$. Figures 3.7(c) and 3.7(d) show that we are unnable to recover λ and the true length-scale process. Moreover, the overestimation of the true correlation structure results in an inflated noise variance.

3.5 Computational burden

Similar to standard GP models, the computational complexity of 2-level GPs scales $\mathcal{O}(N^3)$ in time and $\mathcal{O}(N^2)$ in memory. Numerous alternatives to speed up calculations when using GP priors have been suggested in the literature (Williams

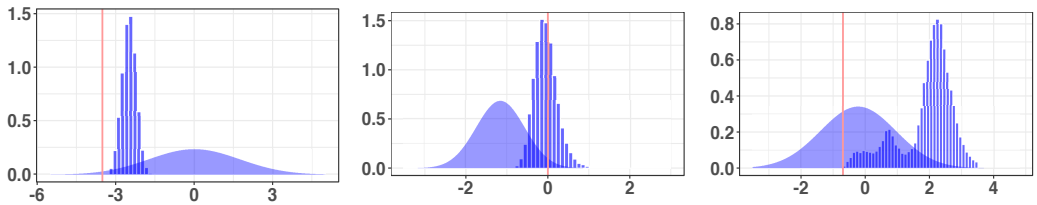


(a) $\log \sigma_\varepsilon^2$ (b) $\log \tau_z^2$ (c) $\log \lambda$

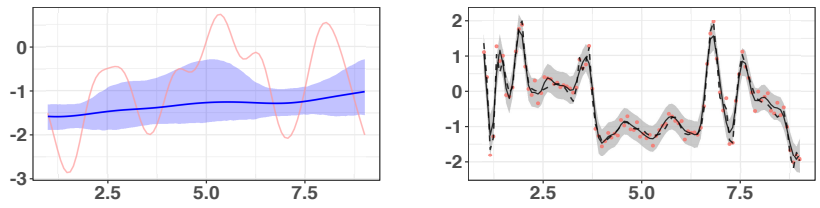


(d) u_5 (e) u_{20} (f) u_{52}

Figure 3.6: Traceplots with cumulative averages and the true parameter value depicted in red.



(a) $\log \sigma_\varepsilon^2$ (b) $\log \tau_z^2$ (c) $\log \lambda$



(d) $\boldsymbol{u}$ (e) $\boldsymbol{z}$

Figure 3.7: Arbitrarily fixing the hyperparameters. (a)-(c) Prior and posterior densities for $\log \sigma_\varepsilon^2$, $\log \tau_z^2$ and $\log \lambda$. Histograms depict samples from the posterior distribution, density plot shows the empirical prior employed and red line corresponds to the true parameter value. (d) The spatially varying parameter. The red line depicts the true parameter and the purple line its estimated posterior mean with 95% credible intervals. (e) The non-stationary function. True function shown with a dashed line with observed data in red versus posterior estimate of $\boldsymbol{z}$ with 95% credible intervals in grey.

and Seeger, 2001; Lawrence et al., 2003; Snelson and Ghahramani, 2006; Titsias, 2009; Moore and Russell, 2015; Datta et al., 2016). A recent review of approaches with a focus on regression tasks is provided by Liu et al. (2018).

Due to the aforementioned computational constraint of the model, special care must be taken during the implementation of expensive matrix algebra calculations. Such operations include matrix inversions, determinant computations, and matrix factorisations. Moreover, numerical issues can arise, resulting for instance, in negative predictive variances.

Avoiding wasteful computations and implementing efficient matrix algebra is particularly crucial for feasible MCMC inference in 2-level GPs. Recently, Finley et al. (2019) and Durrande et al. (2019) discuss efficient computations for GPs models that employ sparse matices. More generally, relevant literature for matrix algebra beyond a sparsity assumption is provided by Golub and Van Loan (1996); Harville (1997), and Press et al. (1988).

3.6 Discussion

This chapter highlights the difficulties of fully Bayesian inference in 2-level GP models and provides initial guidance on how to tackle some of these challenges. First, to infer the spatially varying parameters (or a re-parametrisation of it), we proposed to employ Ell-SS. This method is appealing for our model as it is simple to implement, does not need parameter tuning, and in contrast to other state-of-the-art samplers, such as HMC, does not require derivative information. Second, the lack of identifiability of the covariance hyperparameters prevents efficient parameter estimation. Arbitrarily fixing the hyperparameters can significantly affect inference, especially for the spatially varying parameter that provides information about the correlation structure of the process. Indeed, interpretability of this parameters is one of the key benefits of our model over other non-stationary or deep constructions. To tackle this, we suggested an approach to set empirical priors and to fix some of the parameters in the model. Our studies showed that combining a whitening reparametetrisation with an empirical prior permits effective parameter recovery. Third, efficiently extending the model to higher dimensions is difficult. While the LDL^T representation of the kernel matrix parameters provides appealing information about the correlation structure of the field, its applicability is limited to low-dimensional settings with a moderate number of observations. This is because for a D-dimensional problem, the number of processes needed to parametrise $\Sigma(\cdot)$ with the LDL^T factorisation is $D + D(D-1)/2$. Moreover, in this case, allocating

prior information for the second level hyperparameters becomes more difficult. Finally, practical implementation of 2-level GPs can be hindered by its computational complexity, and during implementation, one must avoid inefficient matrix algebra and care must be taken with numerical instabilities.

CHAPTER 4

FAST BAYESIAN INFERENCE THROUGH AN SPDE FORMULATION

This chapter presents a novel framework to do fully Bayesian inference in 2-level Gaussian process (GP) models. The model is formulated employing Gaussian Markov random fields (GMRFs) with stochastic spatially varying parameters. Importantly, this allows for non-stationarity while also addressing the computational burden through a sparse representation of the precision matrix. The prior field is chosen to be Matérn, and two hyperpriors, for the spatially varying parameters, are considered. One hyperprior is the Ornstein-Uhlenbeck, formulated through an autoregressive process. The other corresponds to the widely used squared exponential. We develop and compare three adaptive Markov chain Monte Carlo (MCMC) schemes and make use of banded matrix operations for faster inference. Furthermore, a novel extension to multi-dimensional settings is proposed through an additive structure that retains the flexibility and scalability of the model, while also inheriting interpretability from the additive approach. A thorough assessment of the efficiency and accuracy of the methods in non-stationary settings is presented for both simulated experiments and a computer emulation problem.

This chapter is the result of collaborative work with Dr Lassi Roininen, Dr Sara Wade, Dr Theo Damoulas, and Prof. Mark Girolami. The work was submitted for publication to Computational Statistics & Data Analysis (under revision).

4.1 Introduction

The stochastic partial differential equation (SPDE) approach introduced by Lindgren et al. (2011) employs Gaussian Markov random fields (GMRFs) to ame-

liorate the computational burden of working with GPs and incorporates a non-stationary framework through spatially varying parameters that are modelled as a linear combination of basis functions. This is similar to the model described in Chapter 2 (Section 2.3), introduced by Paciorek (2003), where a family of closed-form non-stationary covariance functions employs a second latent GP prior to model spatially varying parameters. As previously discussed (see Chapter 3), while these hierarchical constructions are flexible, doing inference in a fully Bayesian framework becomes impractical due to computational demands. Moreover, standard MCMC procedures require careful parameter tuning, exhibit mixing difficulties and require long runs to reach convergence (Paciorek and Schervish, 2006; Neto et al., 2014).

This chapter extends the SPDE formulation of non-stationary GPs considered by Roininen et al. (2019). The model is analogous to SPDE-based constructions in spatial interpolation (Fuglstad et al., 2015; Yue et al., 2014; Fuglstad et al., 2015), and to the non-stationary framework of Paciorek and Schervish (2006), where the spatially varying parameters are modelled as random objects. We incorporate and account for uncertainty in the measurement noise variance and hyperprior parameters and consider two hyperpriors for the spatially varying length-scale to account for different smoothness assumptions.

Moreover, we introduce and offer a comparative evaluation of three MCMC sampling schemes, which are all free of parameter tuning. The first corresponds to an adaptive Metropolis-within-Gibbs (MWG) scheme. The second employs elliptical slice sampling (Ell-SS) combined with re-parametrisations for decoupling the prior, hyperprior, and hyperparameters. The third is a marginal sampler with Ell-SS for a re-parametrised length-scale process. The developed methodology results in a non-stationary hierarchical construction that retains the flexibility of the model introduced by Paciorek and Schervish (2006) but is computationally more efficient, due to the sparsity in the finite-dimensional approximation of the precision matrix.

The sparse 2-level models studied here naturally extend to multiple levels to construct the deep GP models of Dunlop et al. (2018). These hierarchical constructions provide an interpretable structure for non-stationary problems, as well as a sparse framework to address the computational burden, providing a promising route to deeper constructions.

Finally, extensions of the 2-level GPs to multi-dimensional settings are important and necessary in many applications. Existing approaches for two-dimensional settings are based on heavily parametrised models using spectral decompositions (Neto et al., 2014; Paciorek and Schervish, 2006; Risser and Calder, 2017), basis function representations (Katzfuss, 2013), or an isotropic assumption (Heinonen et al., 2016;

Roininen et al., 2019). Instead, we propose a novel extension based on additive Gaussian processes (AGPs) (Duvenaud et al., 2011), that decomposes the function of interest in terms of low-dimensional functions, which are modelled as separable non-stationary processes. Important advantages include increased intrepretability and robustness to curse of dimensionality, while inheriting the appealing flexibility of 2-level GPs. The additive structure permits scalability, by taking advantage of the sparse banded precision matrices, low-dimensional representation, and efficient Kronecker algebra for the separable interaction terms. Moreover, it can capture long-range structures in the data. The choice of interaction terms may be application driven, and hyperpriors can be employed to determine their importance. In this case, the MCMC schemes can be extended through a Gibbs sampling framework. This extension provide an efficient method for data-dense problems in low dimensions but also enables using the construction for multi-dimensional (nD) problems with relatively sparse data, similar to Volodina and Williamson (2018).

The rest of this chapter is organised as follows. We start in Section 4.2 by summarising related work to provide the connection between the work of Paciorek and Schervish (2006) and the SPDE formulation of GPs (Lindgren et al., 2011; Roininen et al., 2019). In Section 4.3, we present the sparse 2-level model and introduce two hyperpriors to model the length-scale. Section 4.4 discusses the proposed sampling schemes employed for one-dimensional problems. Section 4.5 demonstrates how to extend our model to higher dimensional settings while retaining its computational benefits. The experiments in Section 4.6 present a complete empirical evaluation, with a study of the discretisation and sample size effects and performance for different signal types, as well as a comparison with alternative GP models. Finally, Section 4.6.4 applies the methodology to a real-world problem of a NASA rocket booster vehicle.

4.2 Related work and background

Let us recall the 2-level Gaussian process regression (GPR) model for one-dimensional settings introduced in Chapter 3,

$$\begin{aligned} y_n &\sim \mathrm{N}(z(x_n), \sigma_\varepsilon^2), \quad n = 1, \ldots, N, \\ z(\cdot) &\sim \mathrm{GP}\left(0, C_{\boldsymbol{\phi}}^{\mathrm{NS}}(\cdot,\cdot)\right), \\ u := \log \ell(\cdot) &\sim \mathrm{GP}\left(\mu_u, C_{\boldsymbol{\varphi}}^{\mathrm{S}}(\cdot,\cdot)\right), \\ (\tau^2, \boldsymbol{\varphi}, \sigma_\varepsilon^2, \mu_u) &\sim \pi(\tau^2)\pi(\boldsymbol{\varphi})\pi(\sigma_\varepsilon^2)\pi(\mu_u), \end{aligned} \tag{4.1}$$

where

$$C_{\boldsymbol{\phi}}^{\mathrm{NS}}(x_i, x_j) = \frac{\tau^2 2^{1-\nu} \ell(x_i)^{\frac{1}{2}} \ell(x_j)^{\frac{1}{2}}}{\Gamma(\nu) \left([\ell^2(x_i) + \ell^2(x_j)]/2 \right)^{\frac{1}{2}}} \left(2\sqrt{\nu G_{ij}} \right)^{\nu} \mathcal{K}_{\nu} \left(2\sqrt{\nu G_{ij}} \right), \tag{4.2}$$

with $G_{ij} = 2(x_i - x_j)^2/(\ell^2(x_i) + \ell^2(x_j))$, and $C_{\boldsymbol{\varphi}}^{\mathrm{S}}(\cdot, \cdot)$ a stationary covariance function with parameters $\boldsymbol{\varphi}$. As before, the prior for the spatially varying length-scale is assigned over a transformed parameter, defined as $u(\cdot) := \log \ell(\cdot)$, with μ_u representing the a priori constant mean of the log length-scale process.

4.2.1 SPDE formulation of Matérn fields

A Gaussian Markov random field (GMRF) is a random vector $\boldsymbol{z} = (z_1, \dots z_N)^{\mathrm{T}}$, which is Gaussian distributed, $\boldsymbol{z} \sim \mathrm{N}(\boldsymbol{\mu}, Q^{-1})$, and has Markov property: for some $n \neq j, z_n \perp z_j \mid \boldsymbol{z}_{-nj}$, where $\perp$ denotes conditional independence and $\boldsymbol{z}_{-nj}$ refers to all elements in $\boldsymbol{z}$ except for the entries n and j. The Markov property is encoded in the matrix Q, such that $z_n \perp z_j \mid \boldsymbol{z}_{-nj} \iff Q_{nj} = 0$ (Rue and Held, 2005).

Rozanov (1977) proved the general result that if the Fourier transform of the covariance function in a stationary GP has the form $R(\omega) = 1/P(\omega)$, where $P(\omega)$ is a positive, symmetric polynomial, then the process is Markovian. More recently, Lindgren et al. (2011) showed that GMRFs can be presented equivalently as stochastic partial differential equations. By fixing the smoothness parameter $\nu = 2 - D/2$, a GP with stationary Matérn covariance

$$C^{\mathrm{S}}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \tau^2 \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\frac{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|}{\lambda} \right)^{\nu} K_{\nu} \left(\frac{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|}{\lambda} \right), \tag{4.3}$$

and a Markov property can be defined through

$$\left(1 - \lambda^2 \Delta\right) z = \tau \sqrt{\lambda^D} w, \tag{4.4}$$

where $\Delta := \sum_{d=1}^{D} \partial^2/\partial x_d^2$ is the Laplace operator, w is white noise on $\mathbb{R}^D$, and $\mathrm{Var}(w) = \Gamma(\nu + D/2)(4\pi)^{D/2}/\Gamma(\nu)$.

Analogous to the construction of Paciorek and Schervish (2006) for non-stationary covariance functions with spatially varying length-scales, Roininen et al. (2019) derive a SPDE formulation for non-stationary Matérn fields,

$$\left(1 - \ell(\cdot)^2 \Delta\right) z = \tau \sqrt{\ell(\cdot)^D} w, \tag{4.5}$$

where $\ell(\cdot)$ is a spatially varying length-scale, that is modelled as a log-transformed continuous-parameter GP in the hyperprior in Eq. (4.1). An alternative formulation was proposed by Lindgren et al. (2011, Section 3.2), where spatially varying parameters were modelled through a basis function representation. Such a choice gives computational advantages, through a lower dimensional parameter space. However, this requires selecting the number of basis functions, and the ability to flexibly recover changes in the length-scale strongly depends on this choice.

A finite-dimensional approximation of our continuous-parameter model in Eq. (4.5) can be written in vector-matrix format as $L(\boldsymbol{\ell})\boldsymbol{z} = \mathbf{w}$, where $L(\boldsymbol{\ell})$ is a sparse matrix depending on $\ell_j := \ell(jh)$, with h denoting the discretisation step in a chosen finite difference approximation. This model is constructed in such a way that the finite-dimensional approximation converges to the continuous-parameter model in the discretisation limit $h \to 0$ (for proofs, see Roininen et al., 2019). This property guarantees that irrespective of the choice of h, the posteriors, and hence also the estimators, on different meshes, that are dense enough, are essentially the same.

The SPDE formulation in Eq. (4.5) considers periodic boundary conditions, which can lead to undesirable effects in the edges of the estimators. In order to correct a possible boundary effect, one can add points around the boundary. This domain extension offers also a possible benefit in the sparse structure of $L(\boldsymbol{\ell})$. By construction, the matrix $L(\boldsymbol{\ell})$ is a cyclic tridiagonal matrix, and while the Sherman-Morrison formula can be applied to solve this type of systems efficiently (e.g. Seiler and Seiler, 1989), we can simply neglect the matrix elements in the corners once we have applied domain extension and take advantage of the resulting tridiagonal structure.

We note that employing a GP to model $\ell(\cdot)$ results in a similar construction to that presented in Eq. (4.1). The following sections extend the work of Roininen et al. (2019), by including inference of the measurement noise variance and the length-scale hyperparameter. Additionally, we explore different hyperprior models, discuss MCMC algorithms to do inference with these types of models, and present an efficient way to extend the model to higher dimensions.

4.3 Sparse 2-level GP models

The GPR model in Eq. (2.11) can be rephrased through

$$\boldsymbol{y} = \mathcal{A}z + \boldsymbol{\varepsilon} \approx A\boldsymbol{z} + \boldsymbol{\varepsilon}, \tag{4.6}$$

where $\mathcal{A}$ represents a linear mapping from some function space to a finite-dimensional space $\mathbb{R}^N$. Also, $\boldsymbol{\varepsilon} \in \mathbb{R}^N$ is assumed to be zero-mean Gaussian noise with variance $\sigma_\varepsilon^2 I_N$, and independent of z. For computational reasons, we discretise this equation, such that $\mathcal{A}z \approx A\boldsymbol{z}$, obtaining the right hand side of Eq. (4.6), where $A \in \mathbb{R}^{N\times M}$ is a known matrix and $\boldsymbol{z} \in \mathbb{R}^M$ with $\boldsymbol{z} \sim \mathrm{N}(0, C_{\boldsymbol{u}}^{\mathrm{NS}})$. In this case, through the matrix A, we are able to define the grid resolution of the latent fields. In particular, for more rough processes, we may be interested in finer resolutions, while for smooth functions, a sparse grid may be sufficient to obtain an accurate representation.

Our aim is to decompose the inverse covariance matrix $(C_{\boldsymbol{u}}^{\mathrm{NS}})^{-1} := Q_{\boldsymbol{u}} = L(\boldsymbol{u})^{\mathrm{T}} L(\boldsymbol{u})$, where $L(\boldsymbol{u})$ is a sparse matrix that depends on the log length-scale parameters $\boldsymbol{u} = \log(\boldsymbol{\ell})$. The required decomposition can be achieved employing the SPDE approach from Section 4.2.1. An explicit hierarchical formulation of the model is

$$\begin{aligned} \boldsymbol{y} \mid \boldsymbol{z}, \sigma_\varepsilon^2 &\sim \mathrm{N}(A\boldsymbol{z}, \sigma_\varepsilon^2 I_N), \\ \boldsymbol{z} \mid \boldsymbol{u} &\sim \mathrm{N}\left(0, Q_{\boldsymbol{u}}^{-1}\right), \\ \boldsymbol{u} \mid \lambda &\sim \mathrm{N}\left(\boldsymbol{\mu}_u, C_\lambda\right), \\ (\sigma_\varepsilon^2, \lambda) &\sim \pi(\sigma_\varepsilon^2)\pi(\lambda), \end{aligned} \tag{4.7}$$

where $\boldsymbol{\mu}_u$ denotes an M-dimensional vector with all elements equal to μ_u. Because both the length-scale and magnitude parameters cannot be estimated consistently (Zhang, 2004), we use the observed data to set the magnitude and mean of both the stationary and non-stationary processes to improve identifiability (see Chapter 3, Section 3.4.1). The crucial component of the model is $Q_{\boldsymbol{u}}$, the inverse covariance of the GMRF employed to represent the non-stationary GP. This precision matrix depends on $\boldsymbol{u}$, which is assumed to be a constant-mean GP that describes the spatially varying log length-scale, and λ denotes the length-scale parameter of the covariance function that describes the properties of the log length-scale process. A plate diagram of this model is given in Figure 4.1.

4.3.1 Hyperprior processes

In the following, we discuss different types of hyperpriors for $\boldsymbol{u}$. Notice that we are free to assign an inhomogeneous Matérn field for the log length-scale process, introducing more flexibility to the model and resulting in a sparse 3-level construction. For simplicity, we focus on the 2-level case, when the parameters of the log length-scale process are restricted to be constant along the input space.

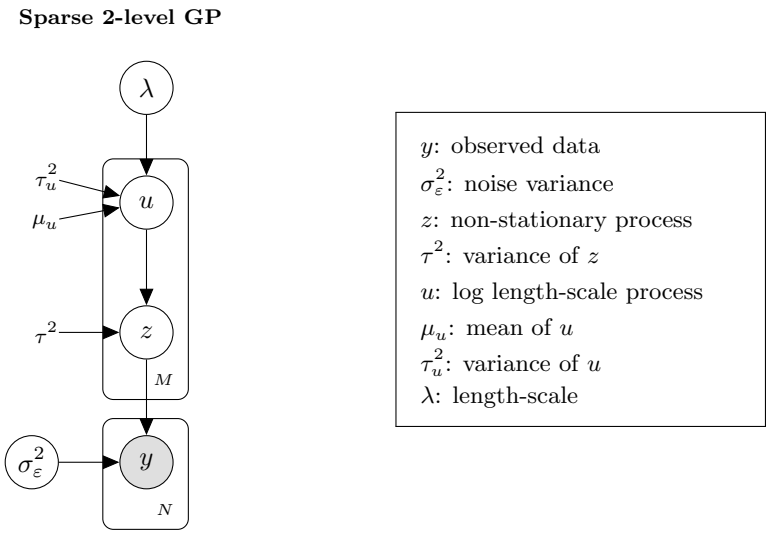


Figure 4.1: Plate diagram for a sparse 2-level GP model. τ^2, τ_u^2 and μ_u are fixed employing the observed data.

Exponential The model requires an hyperprior with sample paths smoother than white noise, otherwise different discretisations of z may affect the posterior estimates (Roininen et al., 2019). One such process is the Ornstein-Uhlenbeck, a member of the stationary Matérn family (Eq. (4.3)) with exponential covariance function obtained by setting $\nu = 1/2$. The Ornstein-Uhlenbeck process has non-differentiable sample paths, allowing quick changes in the behaviour of the log length-scale process. It is the continuous-time counterpart of the first-order autoregressive model AR(1) given by $u_j = \beta u_{j-1} + e_j$ and $e_j \sim \mathrm{N}(0, \sigma^2)$, where u_j is on an uniform lattice $t_j := jh$, $j \in \mathbb{Z}$ with discretisation step h. Without a proof, we note that the AR(1) has an exponential autocovariance for all $\beta > 0$ except for $\beta = 1$ which corresponds to Gaussian random walk, i.e. Brownian motion. While the stable AR(1) requires that $\beta < 1$, this is not a necessary condition here, as our goal is in forming covariance matrices. Let us denote by $a_0 := 1/\sigma$ and $a_1 := \beta/\sigma$. Then, we can construct the inverse of the exponential covariance matrix $(C_\lambda)^{-1} := Q_\lambda = L(\lambda)^{\mathrm{T}} L(\lambda)$, where $L(\lambda)$ is a sparse matrix that depends on λ and τ_u. More precisely, $L(\lambda)$ is a banded matrix, with nonzero elements only on the main diagonal given by $(a_0, \ldots, a_0, 1)$ and the first diagonal above this given by $(a_1, \ldots, a_1)$. The coefficients are defined as

$$a_0 = (\sqrt{h/\lambda} + \sqrt{h/\lambda + 4\lambda/h})/\tau_u\sqrt{8} \text{ and } a_1 = (\sqrt{h/\lambda} - \sqrt{h/\lambda + 4\lambda/h})/\tau_u\sqrt{8}.$$

Hence, we have a sparse representation for the hyperprior precision matrix, and the banded structure in $L(\lambda)$ offers important computational advantages when evaluating $\mathrm{N}(\boldsymbol{u} \mid \boldsymbol{\mu}_u, Q_\lambda^{-1})$, as the required determinant computations, matrix multiplications, and system of equations can be significantly simplified.

Squared exponential In contrast to the AR(1) hyperprior, we have the squared exponential (SE) hyperprior in Eq. (2.19). The SE kernel is recovered when $\nu \to \infty$ in the stationary Matérn covariance. Sample paths from a SE are infinitely differentiable and consequently very smooth. Therefore, when employing a SE hyperprior for the length-scale process, we introduce strong prior smoothness assumptions on how the correlation of the non-stationary process changes with distance. We note that for the SE hyperprior, the precision matrix is dense and therefore, comes at an increased computational cost.

4.4 Inference for one-dimensional problems

In order to efficiently draw samples from the posterior distributions of interest, we explore three MCMC sampling approaches. The first draws samples from the multi-dimensional vector $\boldsymbol{u}$ through an adaptive Metropolis-within-Gibbs (MWG) algorithm. The second employs ancillary augmentation (AA) or whitening (Yu and Meng, 2011) over $\boldsymbol{z}$ and $\boldsymbol{u}$ and uses elliptical slice sampling (Ell-SS) (see Algorithm 4) over the re-parametrised log length-scale process. The third integrates out the non-stationary process, resulting in a marginal sampler that draws from $\boldsymbol{u}$ by combining whitening and Ell-SS to break the correlation between $\boldsymbol{u}$ and λ.

4.4.1 Metropolis-within-Gibbs

This sampling scheme is inspired by that proposed in Roininen et al. (2019) and additionally incorporates adaptive random walks (Roberts and Rosenthal, 2009) for the noise variance, length-scale hyperparameter, and log length-scale process. The procedure is detailed in Algorithm 5.

The MWG framework updates the log length-scale process at each location individually and, regardless of the hyperprior employed, offers computational gains due to the fact that when proposing a single element of the log length-scale process u_m^*, for $m = 1, \ldots, M$, the log-ratio of the prior density of $\boldsymbol{z}$ used in the acceptance

Algorithm 5 Metropolis-within-Gibbs (MWG)

Require: A, $\sigma_\varepsilon^{2^{(0)}}$, $\boldsymbol{u}^{(0)}$, $\boldsymbol{z}^{(0)}$ and $\lambda^{(0)}$

1: **for** $t = 1$ to T **do**

2: Draw: $\log \sigma_\varepsilon^{2^{(t)}}$ using RW-MH$(\log \sigma_\varepsilon^{2^{(t-1)}}, s_1)$ ▷Alg. 2

Step 3 implies computing: $\min\left\{1, \frac{\prod_n \mathrm{N}\left(y_n | A z_n^{(t-1)}, \sigma_\varepsilon^2\right) \pi\left(\log \sigma_\varepsilon^2\right)}{\prod_n \mathrm{N}\left(y_n | A z_n^{(t-1)}, \sigma_\varepsilon^{2^{(t-1)}}\right) \pi\left(\log \sigma_\varepsilon^{2^{(t-1)}}\right)}\right\}$

3: Run Adaptation for s_1

4: Draw: $\boldsymbol{\eta} \sim \mathrm{N}(0, I_{N+M})$

5: Set: $\boldsymbol{z}^{(t)} = \begin{pmatrix} \sigma_\varepsilon^{-1^{(t)}} A \\ L(\boldsymbol{u}^{(t-1)}) \end{pmatrix}^{\dagger} \left(\begin{pmatrix} \sigma_\varepsilon^{-1^{(t)}} \boldsymbol{y} \\ 0 \end{pmatrix} + \boldsymbol{\eta} \right)$ ▷ † Use QR decomposition

6: Draw: $\boldsymbol{u} \sim \mathrm{N}(\boldsymbol{u}^{(t-1)}, P)$ ▷$P = \mathrm{diag}(\sigma_{u_1}^2, \ldots, \sigma_{u_n}^2)$

7: Set: $\boldsymbol{u}' = \boldsymbol{u}^{(t-1)}$ and $\boldsymbol{u}^{(t)} = \boldsymbol{u}^{(t-1)}$

8: **for** $m = 1$ to M **do**

9: Set: $\boldsymbol{u}_{j \neq m} = (u_1^{(t)}, \ldots, u_{m-1}^{(t)}, u_m, u_{m+1}^{(t-1)}, \ldots, u_M^{(t-1)})^T$

10: Compute: $\alpha_{u_m} = \min\left\{1, \frac{\mathrm{N}\left(\boldsymbol{z}^{(t)} | 0, C_{\boldsymbol{u}_{j \neq m}}\right) \mathrm{N}\left(\boldsymbol{u}_{j \neq m} | \boldsymbol{\mu}_u, C_\lambda^{(t-1)}\right)}{\mathrm{N}\left(\boldsymbol{z}^{(t)} | 0, C_{\boldsymbol{u}'}\right) \mathrm{N}\left(\boldsymbol{u}' | \boldsymbol{\mu}_u, C_\lambda^{(t-1)}\right)}\right\}$

11: With probability α_{u_m} set $u_m^{(t)} = u_m$ and $u_m' = u_m$; otherwise set $u_m^{(t)} = u_m^{(t-1)}$

12: **end for**

13: Run Adaptation for P

14: Draw: $\log \lambda | \log \lambda^{(t-1)}$ using RW-MH$(\log \lambda^{(t-1)}, s_2)$ ▷Alg. 2

Step 3 implies computing: $\min\left\{1, \frac{\mathrm{N}\left(\boldsymbol{u}^{(t)} | \boldsymbol{\mu}_u, C_\lambda\right) \pi(\log \lambda)}{\mathrm{N}\left(\boldsymbol{u}^{(t)} | \boldsymbol{\mu}_u, C_{\lambda^{(t-1)}}\right) \pi\left(\log \lambda^{(t-1)}\right)}\right\}$

15: Run Adaptation for s_2

16: **end for**

probability simplifies to

$$\log\left(\frac{\mathrm{N}(\boldsymbol{z} \mid 0, Q_{\boldsymbol{u}^*}^{-1})}{\mathrm{N}(\boldsymbol{z} \mid 0, Q_{\boldsymbol{u}}^{-1})}\right) = \log\det(L(\boldsymbol{u}^*)L(\boldsymbol{u})^{-1}) - \frac{1}{2}\boldsymbol{z}^{\mathrm{T}}\left(L(\boldsymbol{u}^*)^{\mathrm{T}}L(\boldsymbol{u}^*) - L(\boldsymbol{u})^{\mathrm{T}}L(\boldsymbol{u})\right)\boldsymbol{z}.$$

Here $\boldsymbol{u}^*$ is the proposed log length-scale vector, obtained by updating the m^{th} element of $\boldsymbol{u}$ to u_m^*, and combined with pentadiagonal form of the precision matrix, resulting from multiplication of tridiagonal matrices $Q_{\boldsymbol{u}} = L(\boldsymbol{u})^{\mathrm{T}}L(\boldsymbol{u})$, the computational complexity of the quadratic term in the log-ratio is reduced from $\mathcal{O}(M^2)$ to $\mathcal{O}(1)$. Moreover, the log-determinant can be computed through numerically stable and inexpensive operations; for details, see Roininen et al. (2019, Section 6). Similarly,

the log-ratio of the prior density of $\boldsymbol{u}$ simplifies to

$$\log\left(\frac{\mathrm{N}(\boldsymbol{u}^* \mid \boldsymbol{\mu}_u, C_\lambda)}{\mathrm{N}(\boldsymbol{u} \mid \boldsymbol{\mu}_u, C_\lambda)}\right) = -\frac{1}{2}\left([(u_m^*)^2 - u_m^2]Q_{\lambda\, mm} + \sum_{j\neq m}[u_m^* - u_m]u_j Q_{\lambda\, mj}\right),$$

where $Q_{\lambda\, mj}$ denotes the (m, j) element of the matrix Q_λ. Further computational gains are possible when we employ the AR(1) hyperprior, as the tridiagonal form of $Q_\lambda = L(\lambda)^{\mathrm{T}}L(\lambda)$, resulting from the sparse AR(1) construction of $L(\lambda)$, reduces this operation from $\mathcal{O}(M)$ to $\mathcal{O}(1)$.

Additionally, when proposing a new hyperparameter λ^*, we must evaluate

$$\log\left(\frac{\mathrm{N}(\boldsymbol{u} \mid \boldsymbol{\mu}_u, C_{\lambda^*})}{\mathrm{N}(\boldsymbol{u} \mid \boldsymbol{\mu}_u, C_\lambda)}\right) = \frac{1}{2}\log\det(Q_{\lambda^*}Q_\lambda^{-1}) - \frac{1}{2}(\boldsymbol{u} - \boldsymbol{\mu}_u)^{\mathrm{T}}(Q_\lambda - Q_{\lambda^*})(\boldsymbol{u} - \boldsymbol{\mu}_u).$$

For the SE hyperprior, this requires the inversion of a dense $M \times M$ matrix, while the tridiagonal form of Q_λ for the AR(1) hyperprior makes this considerably cheaper by reducing the computational complexity of this log-ratio term from $\mathcal{O}(M^3)$ to $\mathcal{O}(M)$. In addition, our simulation studies show that this algorithm does not perform well when the hyperprior for $u(\cdot)$ has strong smoothness assumptions, such as those induced by employing a SE covariance function. This flaw motives us to explore alternative algorithms.

4.4.2 Whitened elliptical slice sampling

Whitened elliptical slice sampling (w-Ell-SS) combines Ell-SS, outlined in Algorithm 4, with whitening to break the correlation between the prior and its corresponding hyperparameters. We can equivalently define the unknown function as $\boldsymbol{z} = L(\boldsymbol{u})^{-1}\boldsymbol{\xi}$ with $\boldsymbol{\xi} \sim \mathrm{N}(0, I_M)$ and the log length-scale vector as $\boldsymbol{u} = R_\lambda\boldsymbol{\zeta} + \boldsymbol{\mu}_u$ with $\boldsymbol{\zeta} \sim \mathrm{N}(0, I_M)$. For the AR(1) hyperprior, $R_\lambda := L(\lambda)^{-1}$; whereas, for the SE hyperprior, we define R_λ to be the lower-triangular Cholesky factor of C_λ. Reparametrising in terms of the whitened parameters $\boldsymbol{\xi}$ and $\boldsymbol{\zeta}$, results in the joint posterior,

$$\pi(\boldsymbol{\zeta}, \boldsymbol{\xi}, \lambda, \sigma_\varepsilon^2 \mid \boldsymbol{y}) \propto \mathrm{N}(\boldsymbol{y} \mid AL(R_\lambda\boldsymbol{\zeta} + \boldsymbol{\mu}_u)^{-1}\boldsymbol{\xi}, \sigma_\varepsilon^2 I_N)\mathrm{N}(\boldsymbol{\xi} \mid 0, I_M)\mathrm{N}(\boldsymbol{\zeta} \mid 0, I_M)\pi(\lambda)\pi(\sigma_\varepsilon^2).$$

The sampling method is detailed in Algorithm 6. As opposed to the MWG, the log length scales, $\boldsymbol{u}$, are updated jointly through the whitened parameter $\boldsymbol{\zeta}$. In this case, the likelihood can be evaluated as a product of univariate Gaussian distributions, after computing $\boldsymbol{u} = R_\lambda\boldsymbol{\zeta} + \boldsymbol{\mu}_u$ and solving $L(\boldsymbol{u})\boldsymbol{z} = \boldsymbol{\xi}$. Regardless of

Algorithm 6 Whitened elliptical slice sampling (w-Ell-SS)

Require: A, $\sigma_\varepsilon^{2(0)}$, $\boldsymbol{\zeta}^{(0)}$, $\boldsymbol{\xi}^{(0)}$, $\lambda^{(0)}$, $\boldsymbol{u} = R_{\lambda^{(0)}}\boldsymbol{\zeta}^{(0)} + \boldsymbol{\mu}_u$ and $\boldsymbol{z} = L(\boldsymbol{u})^{-1}\boldsymbol{\xi}^{(0)}$

1: **for** $t = 1$ to T **do**

2: Draw: $\log \sigma_\varepsilon^{2(t)}$ using RW-MH($\log \sigma_\varepsilon^{2(t-1)}, s_1$) ▷Alg. 2

Step 3 implies computing: $\min\left\{1, \frac{\prod_n \mathrm{N}\left(y_n | Az_n, \sigma_\varepsilon^2\right)\pi\left(\log \sigma_\varepsilon^2\right)}{\prod_n \mathrm{N}\left(y_n | Az_n, \sigma_\varepsilon^{2(t-1)}\right)\pi\left(\log \sigma_\varepsilon^{2(t-1)}\right)}\right\}$

3: Run Adaptation for s_1

4: Draw: $\boldsymbol{\zeta}^{(t)}$ using Ell-SS($\boldsymbol{\zeta}^{(t-1)}, I_M$) ▷Alg. 4

Evaluation of log likelihood in Step 11 implies:

a. Updating: $\boldsymbol{u}' = R_{\lambda^{(t-1)}}\boldsymbol{\zeta}' + \boldsymbol{\mu}_u$

b. Solving: $L(\boldsymbol{u}')\boldsymbol{z}' = \boldsymbol{\xi}^{(t-1)}$

c. Computing $\log \prod_n \mathrm{N}(y_n \mid Az'_n, \sigma_\varepsilon^{2(t)})$

5: Draw: $\log \lambda^{(t)}$ using RW-MH($\log \lambda^{(t-1)}, s_2$) ▷Alg. 2

Step 3 implies:

a. Computing: $R_{\lambda'}$

b. Updating: $\boldsymbol{u}' = R_{\lambda'}\boldsymbol{\zeta}^{(t)} + \boldsymbol{\mu}_u$

c. Solving: $L(\boldsymbol{u}')\boldsymbol{z}' = \boldsymbol{\xi}^{(t-1)}$

d. Computing: $\min\left\{1, \frac{\prod_n \mathrm{N}\left(y_n | Az'_n, \sigma_\varepsilon^{2(t)}\right)\pi\left(\log \lambda'\right)}{\prod_n \mathrm{N}\left(y_n | Az_n, \sigma_\varepsilon^{2(t)}\right)\pi\left(\log \lambda^{(t-1)}\right)}\right\}$

6: Run Adaptation for s_2

7: Draw: $\boldsymbol{\eta} \sim \mathrm{N}(0, I_{N+M})$

8: Set: $\boldsymbol{z} = \begin{pmatrix} \sigma_\varepsilon^{-1(t)} A \\ L(\boldsymbol{u}) \end{pmatrix}^{\dagger} \left(\begin{pmatrix} \sigma_\varepsilon^{-1(t)} \boldsymbol{y} \\ 0 \end{pmatrix} + \boldsymbol{\eta} \right)$ ▷ † Use QR decomposition

9: Solve: $L(\boldsymbol{u})\boldsymbol{\xi}^{(t)} = \boldsymbol{z}$

10: **end for**

the hyperprior employed, the latter system of equations $L(\boldsymbol{u})\boldsymbol{z} = \boldsymbol{\xi}$ can be solved in $\mathcal{O}(M)$ operations by taking advantage of the tridiagonal structure of $L(\boldsymbol{u})$ (Rue and Held, 2005). The former system of equations $\boldsymbol{u} = R_\lambda\boldsymbol{\zeta} + \boldsymbol{\mu}_u$ requires matrix multiplication, resulting in $\mathcal{O}(M^2)$ operations; however, for the AR(1) hyperprior, we can equivalently solve $L(\lambda)(\boldsymbol{u} - \boldsymbol{\mu}_u) = \boldsymbol{\zeta}$ and make use of the banded form of $L(\lambda)$ to reduce this to $\mathcal{O}(M)$ operations.

Thus, while MWG requires looping over the elements of the M-dimensional log length-scale vector, with each operation costing $\mathcal{O}(1)$ operations for the AR(1) hyperprior and $\mathcal{O}(M)$ operations for the SE hyperprior, the w-Ell-SS instead updates this vector jointly through $\mathcal{O}(M)$ for the AR(1) hyperprior and $\mathcal{O}(M^2)$ operations

for the SE hyperprior. However, as Ell-SS is a rejection free sampling method, each iteration may require several likelihood evaluations, mitigating any gain in computation time of this scheme.

4.4.3 Marginal elliptical slice sampling

In simulation studies, we found that integrating out the unknown function $\boldsymbol{z}$ significantly improves the mixing of $\boldsymbol{u}$ and its hyperparameters. The log marginal likelihood of the data corresponds to

$$\log \pi(\boldsymbol{y} \mid \boldsymbol{u}, \lambda, \sigma_\varepsilon^2) = -\frac{N}{2}\log(2\pi) - \frac{1}{2}\log\det(\Psi) - \frac{1}{2}\boldsymbol{y}^{\mathrm{T}}\Psi^{-1}\boldsymbol{y}, \tag{4.8}$$

where $\Psi = AQ_{\boldsymbol{u}}^{-1}A^{\mathrm{T}} + \sigma_\varepsilon^2 I_N$. Again, we use whitening to decouple $\boldsymbol{u}$ and λ, with the re-parametrisation $\boldsymbol{\zeta} = R_\lambda^{-1}(\boldsymbol{u} - \boldsymbol{\mu}_u)$ and $R_\lambda = L(\lambda)^{-1}$ for the AR(1) hyperprior or $R_\lambda = \mathrm{chol}(C_\lambda)$ for the SE hyperprior. The posterior is

$$\pi(\boldsymbol{\zeta}, \lambda, \sigma_\varepsilon^2 \mid \boldsymbol{y}) \propto \mathrm{N}(\boldsymbol{y} \mid 0, AQ^{-1}_{R_\lambda\boldsymbol{\zeta}+\boldsymbol{\mu}_u}A^{\mathrm{T}} + \sigma_\varepsilon^2 I_N)\mathrm{N}(\boldsymbol{\zeta} \mid 0, I_M)\pi(\lambda)\pi(\sigma_\varepsilon^2).$$

Marginal elliptical slice sampling (m-Ell-SS) is detailed in Algorithm 7. Again, the log length scales $\boldsymbol{u}$ are updated jointly through the whitened parameter $\boldsymbol{\zeta}$. This requires first computing $\boldsymbol{u} = R_\lambda\boldsymbol{\zeta} + \boldsymbol{\mu}_u$, an $\mathcal{O}(M)$ operation for the AR(1) hyperprior and $\mathcal{O}(M^2)$ operation for the SE hyperprior. However, in comparison with the w-Ell-SS, which proceeds by solving $L(\boldsymbol{u})\boldsymbol{z} = \boldsymbol{\xi}$ and simply taking the product of univariate Gaussians in $\mathcal{O}(M)$ operations, we must evaluate the marginal likelihood in Eq. (4.8).

When computing the marginal likelihood, we emphasise that the required calculations for Ψ can be computed employing the Woodbury identity (see Apendix A);

$$\Psi^{-1} = \sigma_\varepsilon^{-2}\left(I_N - \sigma_\varepsilon^{-2}A\left(L(\boldsymbol{u})^{\mathrm{T}}L(\boldsymbol{u}) + \sigma_\varepsilon^{-2}A^{\mathrm{T}}A\right)^{-1}A^{\mathrm{T}}\right).$$

While this identity also requires a matrix inversion, note that $L(\boldsymbol{u})^{\mathrm{T}}L(\boldsymbol{u}) + \sigma_\varepsilon^{-2}A^{\mathrm{T}}A$ is also banded and therefore computations are considerably cheaper. Indeed, the quadratic term in the marginal likelihood in Eq. (4.8) is

$$\sigma_\varepsilon^{-2}\left(\boldsymbol{y}^{\mathrm{T}}\boldsymbol{y} - \sigma_\varepsilon^{-2}\boldsymbol{y}^{\mathrm{T}}A\left(L(\boldsymbol{u})^{\mathrm{T}}L(\boldsymbol{u}) + \sigma_\varepsilon^{-2}A^{\mathrm{T}}A\right)^{-1}A^{\mathrm{T}}\boldsymbol{y}\right),$$

with the most expensive operation of order $\mathcal{O}(M)$. Specifically, the first term $\boldsymbol{y}^{\mathrm{T}}\boldsymbol{y}$ can be computed in $\mathcal{O}(N)$ operations, while the second term can be efficiently com-

Algorithm 7 Marginal elliptical slice sampling (m-Ell-SS)

Require: A, $\sigma_\varepsilon^{2(0)}$, $\boldsymbol{\zeta}^{(0)}$, $\lambda^{(0)}$, and $\boldsymbol{u} = R_{\lambda^{(0)}}\boldsymbol{\zeta}^{(0)} + \boldsymbol{\mu}_u$

1: **for** $t = 1$ to T **do**

2: Draw: $\log \sigma_\varepsilon^{2(t)}$ using RW-MH($\log \sigma_\varepsilon^{2(t-1)}, s_1$) ▷Alg. 2

Step 3 implies computing: $\min\left\{1, \frac{\mathrm{N}\left(\boldsymbol{y}|0, AQ_{\boldsymbol{u}}^{-1}A^{\mathrm{T}} + \sigma_\varepsilon^2 I_N\right)\pi\left(\log \sigma_\varepsilon^2\right)}{\mathrm{N}\left(\boldsymbol{y}|0, AQ_{\boldsymbol{u}}^{-1}A^{\mathrm{T}} + \sigma_\varepsilon^{2(t-1)} I_N\right)\pi\left(\log \sigma_\varepsilon^{2(t-1)}\right)}\right\}$

3: Run Adaptation for s_1

4: Draw: $\boldsymbol{\zeta}^{(t)}$ using Ell-SS($\boldsymbol{\zeta}^{(t-1)}, I_M$) ▷Alg. 4

Evaluation of log likelihood in Step 11 implies:

a. Computing: $\boldsymbol{u}' = R_{\lambda^{(t-1)}}\boldsymbol{\zeta}' + \boldsymbol{\mu}_u$

b. Computing: $\log \mathrm{N}(\boldsymbol{y} \mid 0, AQ_{\boldsymbol{u}'}^{-1}A^{\mathrm{T}} + \sigma_\varepsilon^{2(t)} I_N)$

5: Draw: $\log \lambda^{(t)}$ using RW-MH($\log \lambda^{(t-1)}, s_2$) ▷Alg. 2

Step 3 implies:

a. Computing: $R_{\lambda'}$

b. Updating: $\boldsymbol{u}' = R_{\lambda'}\boldsymbol{\zeta}^{(t)} + \boldsymbol{\mu}_u$

c. Computing: $\min\left\{1, \frac{\mathrm{N}\left(\boldsymbol{y}|0, AQ_{\boldsymbol{u}'}^{-1}A^{\mathrm{T}} + \sigma_\varepsilon^{2(t)} I_N\right)\pi\left(\log \lambda'\right)}{\mathrm{N}\left(\boldsymbol{y}|0, AQ_{\boldsymbol{u}}^{-1}A^{\mathrm{T}} + \sigma_\varepsilon^{2(t)} I_N\right)\pi\left(\log \lambda^{(t-1)}\right)}\right\}$

6: Run Adaptation for s_2

7: **end for**

puted by breaking it into three separate operations. First, we set $\boldsymbol{\varsigma} = A^{\mathrm{T}}\boldsymbol{y}$, with computational complexity reduced from $\mathcal{O}(MN)$ to $\mathcal{O}(M)$ through sparsity in A. Next, we solve $(L(\boldsymbol{u})^{\mathrm{T}}L(\boldsymbol{u}) + \sigma_\varepsilon^{-2}A^{\mathrm{T}}A)\boldsymbol{\varrho} = \boldsymbol{\varsigma}$ in $\mathcal{O}(M)$ operations due to the banded form of the matrix. Finally, we compute $\boldsymbol{\varsigma}^{\mathrm{T}}\boldsymbol{\varrho}$, with a cost of $\mathcal{O}(M)$ operations. Computing the determinant, on the other hand, is more expensive with the dominant term costing $\mathcal{O}(N^3)$ or $\mathcal{O}(MN)$, whichever is greater. Specifically, we must first solve $(L(\boldsymbol{u})^{\mathrm{T}}L(\boldsymbol{u}) + \sigma_\varepsilon^{-2}A^{\mathrm{T}}A)B = A^{\mathrm{T}}$, with complexity $\mathcal{O}(MN)$, and then compute AB, with reduced complexity $\mathcal{O}(MN)$ due to sparsity in A. Finally, the determinant of the $N \times N$ matrix Ψ^{-1} is computed.

In addition, when proposing new values for the noise variance σ_ε^2 or the length scale λ, we must recompute the marginal likelihood in Eq. (4.8), as opposed to evaluating the product of N univariate Gaussians for the w-Ell-SS scheme, increasing the cost of these steps as well. However, in the marginal scheme, in contrast to both MWG and w-Ell-SS, sampling of $\boldsymbol{z}$ is no longer required. We also note the computational gains of the AR(1) over the SE hyperprior deteriorate when the determinant evaluation dominates this computation, i.e. when $N^3 > M^2$.

The increased computational cost of the marginal scheme comes with improved mixing, and this trade-off is examined in the simulation studies of Section 4.6.1. In contrast to MWG, this scheme performs well regardless of the hyperprior employed.

4.5 Extension for D-dimensional problems

To extend the sparse 2-level GP model to higher dimensional settings, while maintaining its computational benefits, we propose a novel construction utilising AGP models (Duvenaud et al., 2011). First, the model is presented, followed by a description of the extended inference procedure.

4.5.1 Sparse non-stationary 2-level additive models

Additive regression models decompose the regression function into main effects and interactions. Linear regression is a classic example, and nonparametric additive models (Friedman and Stuetzle, 1981; Buja et al., 1989) provide increased flexibility, while retaining interpretability and robustness to the input dimension, when compared with general nonparameteric surfaces. The additive GP formulation results from considering the sum and product of covariance functions, two operations for constructing valid covariance functions in D-dimensions. This provides a flexible and interpretable model for the unknown function to include main first-order terms up to D-order interaction terms, assumed to be separable across dimensions.

In an AGP model, the choice between low-order and high-order terms represents a trade-off between interpretability and accuracy. On one hand, by including only first-order terms, the model can capture long-range structures and has increased intrepretability. On the other, including only a D-order separable function increases flexibility and complexity. Duvenaud et al. (2011) include all interaction terms and develop a maximum marginal likelihood approach to determine the importance of each term. Additionally, they propose an efficient algorithm, despite the exponential number of terms, through parametrisations that limit the number of hyperparameters. Interestingly, their experiments show that typically only a few orders of interactions are important. Alternatively, the choice of terms in the AGP may be application driven; more recently, this is the approach taken in Cheng et al. (2019) for longitudinal biomedical data. Another interesting direction in Gilboa et al. (2013) constructs projected additive GPs through first-order functions of linear projections of the inputs.

For notational simplicity, in the following, we focus on the two-dimensional set-

ting, including both the main and interaction terms for generality. The model construction and inference can be applied to D-dimensional settings, through appropriate choice of the terms to include in the additive formulation. In two-dimensional problems, the discretisation is based on a complete $M_1 \times M_2$ grid, with the noisy realisations modelled through

$$\boldsymbol{y} = A_1\boldsymbol{z}_1 + A_2\boldsymbol{z}_2 + A_3\boldsymbol{z}_3 + \boldsymbol{\varepsilon},$$

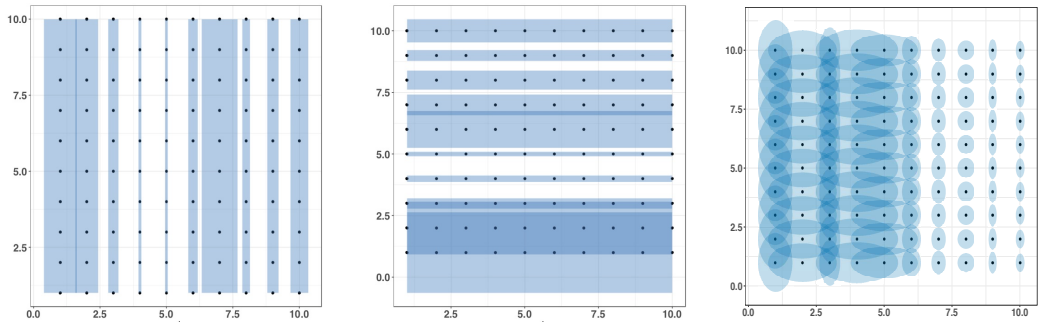
where $A_1 \in \mathbb{R}^{N\times M_1}$, $A_2 \in \mathbb{R}^{N\times M_2}$ and $A_3 \in \mathbb{R}^{N\times(M_1M_2)}$ are known matrices. We assume $z_1(\cdot)$ and $z_2(\cdot)$ are independent one-dimensional non-stationary processes, while $z_3(\cdot)$ is a two-dimensional, separable non-stationary process. Thus, $\boldsymbol{z}_r \in \mathbb{R}^{M_r}$ denotes the vector formed by the first-order non-stationary processes at the M_r locations in dimension $r = 1, 2$, while $\boldsymbol{z}_3 \in \mathbb{R}^{M_1M_2}$ collects the second-order non-stationary process at all locations on the complete $M_1 \times M_2$ grid.

The hierarchical structure of the model (depicted in Figure 4.3) is

$$\begin{aligned}
\boldsymbol{y} \mid \{\boldsymbol{z}_r\}_{r=1}^3, \sigma_\varepsilon^2 &\sim \mathrm{N}(A_1\boldsymbol{z}_1 + A_2\boldsymbol{z}_2 + A_3\boldsymbol{z}_3, \sigma_\varepsilon^2 I_N),\\
\boldsymbol{z}_r \mid \boldsymbol{u}_r &\sim \mathrm{N}\left(0, C^{\mathrm{NS}}_{\boldsymbol{u}_r}\right), \quad r = 1, 2,\\
\boldsymbol{z}_3 \mid \boldsymbol{u}_3, \boldsymbol{u}_4 &\sim \mathrm{N}\left(0, C^{\mathrm{NS}}_{\boldsymbol{u}_3,\boldsymbol{u}_4}\right),\\
\boldsymbol{u}_s \mid \lambda_s &\sim \mathrm{N}\left(\boldsymbol{\mu}_{u_s}, C^{\mathrm{S}}_{\lambda_s}\right), \quad s = 1, 2, 3, 4,\\
(\sigma_\varepsilon^2, \boldsymbol{\lambda}) &\sim \pi(\sigma_\varepsilon^2)\pi(\lambda_1)\pi(\lambda_2)\pi(\lambda_3)\pi(\lambda_4),
\end{aligned} \tag{4.9}$$

with $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_4)$. In Eq. (4.9), we have four one-dimensional length-scale processes: two describing the correlation changes in each direction independently and two incorporating that information in a two-dimensional process, through a separable assumption $C^{\mathrm{NS}}_{\boldsymbol{u}_3,\boldsymbol{u}_4}(\boldsymbol{x}_i, \boldsymbol{x}_j) = C^{\mathrm{NS}}_{\boldsymbol{u}_3}(x_{i1}, x_{j1})C^{\mathrm{NS}}_{\boldsymbol{u}_4}(x_{i2}, x_{j2})$. A visualisation of the non-stationary additive covariance function is provided in Figure 4.2.

Because the AGP is based on one-dimensional kernels, we can directly apply the methodology discussed in Section 4.3 for any of the hyperpriors studied. Instead, a direct extension of the SPDE model to two-dimensional settings will not allow us to employ the AR(1) hyperprior and benefit from its computational advantages. This is because a two-dimensional exponential covariance does not have a valid Markov representation. Furthermore, the additive and hierarchical structure of the model in Eq. (4.9) favours interpretability about the behaviour of the correlation in each dimension.



(a) 1st-order non-stationary covariance $C^{\text{NS}}_{\boldsymbol{u}_1}$

(b) 1st-order non-stationary covariance $C^{\text{NS}}_{\boldsymbol{u}_2}$

(c) 2nd-order non-stationary covariance $C^{\text{NS}}_{\boldsymbol{u}_3,\boldsymbol{u}_4}$

Figure 4.2: The non-stationary additive covariance function in 2-D with main effects and an interaction is the sum of the three terms: C^{NS}=$C^{\text{NS}}_{\boldsymbol{u}_1}$ +$C^{\text{NS}}_{\boldsymbol{u}_2}$ +$C^{\text{NS}}_{\boldsymbol{u}_3,\boldsymbol{u}_4}$. At each location the covariance function will make use the data contained within the shaded region in each of the plots. The 1st-order terms can pool together data across dimensions for long-range correlations, while the 2nd-order terms can capture local behavior in both dimensions.

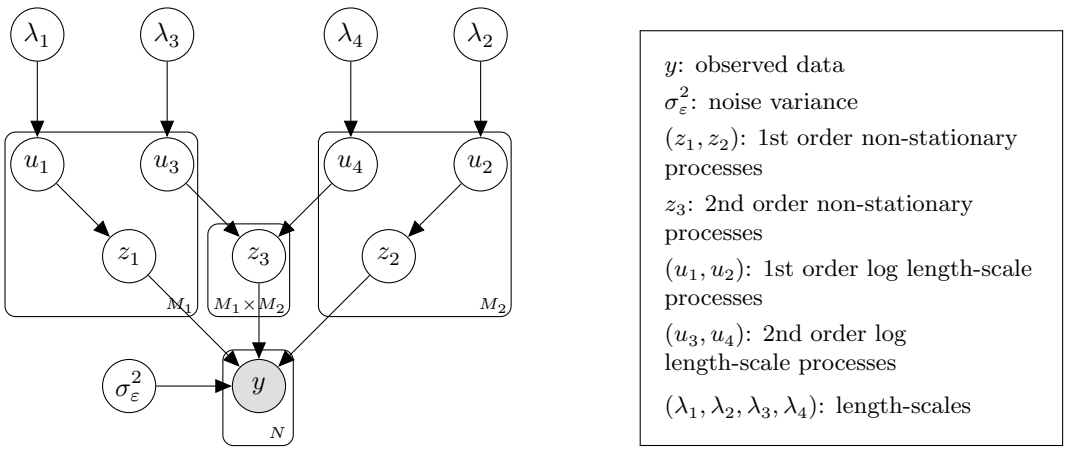


Figure 4.3: Plate diagram for a sparse non-stationary 2-level additive GP model.

4.5.2 Inference for non-stationary 2-level additive models

The posterior for the 2-level additive non-stationary model in Eq. (4.9) is

$$\begin{aligned}\pi(\{\boldsymbol{z}_r\}_{r=1}^3, \{\boldsymbol{u}_s, \lambda_s\}_{s=1}^4, \sigma_\varepsilon^2 \mid \boldsymbol{y}) \propto \mathrm{N}(\boldsymbol{y} \mid A_1\boldsymbol{z}_1 + A_2\boldsymbol{z}_2 + A_3\boldsymbol{z}_3, \sigma_\varepsilon^2 I_N) \\ \mathrm{N}(\boldsymbol{z}_1 \mid 0, Q_{\boldsymbol{u}_1}^{-1})\mathrm{N}(\boldsymbol{z}_2 \mid 0, Q_{\boldsymbol{u}_2}^{-1})\mathrm{N}(\boldsymbol{z}_3 \mid 0, Q_{\boldsymbol{u}_3,\boldsymbol{u}_4}^{-1}) \\ \mathrm{N}(\boldsymbol{u}_1 \mid \boldsymbol{\mu}_{u_1}, C_{\lambda_1}) \cdots \mathrm{N}(\boldsymbol{u}_4 \mid \boldsymbol{\mu}_{u_4}, C_{\lambda_4})\pi(\lambda_1) \cdots \pi(\lambda_4)\pi(\sigma_\varepsilon^2),\end{aligned}$$

with $Q_{\boldsymbol{u}_3,\boldsymbol{u}_4}^{-1}$ being a separable covariance matrix, defined as $Q_{\boldsymbol{u}_{3,4}}^{-1} := Q_{\boldsymbol{u}_3}^{-1} \otimes Q_{\boldsymbol{u}_4}^{-1}$, where $\otimes$ denotes the Kronecker product. The three inference schemes described in Section 4.4 can be appropriately extended through a blocked Gibbs sampler, that updates the three blocks of parameters $(\boldsymbol{z}_1, \boldsymbol{u}_1, \lambda_1)$; $(\boldsymbol{z}_2, \boldsymbol{u}_2, \lambda_2)$; and $(\boldsymbol{z}_3, \boldsymbol{u}_3, \boldsymbol{u}_4, \lambda_3, \lambda_4)$ from their full conditional distributions. Following from the one-dimensional synthetic experiments of Section 4.6.1, we focus on the marginal sampler of Section 4.4.3. We will refer to it as the block marginal elliptical slice sampler (block-m-Ell-SS); in this case, although we are not integrating out the processes $\{\boldsymbol{z}_r\}_{r=1}^3$, we use the marginal likelihood to sample the length-scale process and corresponding length-scale hyperparameters in each block. For instance, when sampling the block $(\boldsymbol{z}_1, \boldsymbol{u}_1, \lambda_1)$, the full conditional factorises as

$$\pi(\boldsymbol{z}_1, \boldsymbol{\zeta}_1, \lambda_1 \mid \boldsymbol{y}, \sigma_\varepsilon^2, \boldsymbol{z}_2, \boldsymbol{z}_3) = \pi(\boldsymbol{\zeta}_1, \lambda_1 \mid \boldsymbol{y}, \sigma_\varepsilon^2, \boldsymbol{z}_2, \boldsymbol{z}_3)\pi(\boldsymbol{z}_1 \mid \boldsymbol{\zeta}_1, \lambda_1, \boldsymbol{y}, \sigma_\varepsilon^2, \boldsymbol{z}_2, \boldsymbol{z}_3),$$

with $\boldsymbol{\zeta}_1 = R_{\lambda_1}^{-1}(\boldsymbol{u}_1 - \boldsymbol{\mu}_{u_1})$ denoting the whitened parameter. Thus, we first sample from the block marginal $\pi(\boldsymbol{\zeta}_1, \lambda_1 \mid \boldsymbol{y}, \sigma_\varepsilon^2, \boldsymbol{z}_2, \boldsymbol{z}_3)$ utilising the steps described in Section 4.4.3, with the marginal likelihood replaced by $\mathrm{N}(\boldsymbol{y} - A_2\boldsymbol{z}_2 - A_3\boldsymbol{z}_3 \mid 0, A_1 Q_{\boldsymbol{u}_1}^{-1} A_1^{\mathrm{T}} + \sigma_\varepsilon^2 I_N)$. The algorithm is detailed in Algorithm 8.

For efficiency in evaluating the block marginal likelihood obtained from integration of $\boldsymbol{z}_r$, $r = 1, 2$, the matrix determinant lemma (Harville, 1997) must be employed to avoid computing the log determinant of an $N \times N$ matrix and instead evaluate,

$$\log\det\left(A_1 Q_{\boldsymbol{u}_1}^{-1} A_1^{\mathrm{T}} + \sigma_\varepsilon^2 I_N\right) = N\log(\sigma_\varepsilon^2) + \log\det\left(Q_{\boldsymbol{u}_1} + \sigma_\varepsilon^2 A_1^{\mathrm{T}} A_1\right) - \log\det(Q_{\boldsymbol{u}_1}).$$

When an interaction term is employed in the model, the algorithm requires samples from the posterior of $\boldsymbol{z}_3$, which is a Gaussian distribution with mean $\boldsymbol{\mu}_{z_3} = \sigma_\varepsilon^{-2}\Sigma_{z_3} A_3^{\mathrm{T}}(\boldsymbol{y} - A_1\boldsymbol{z}_1 - A_2\boldsymbol{z}_2)$ and variance $\Sigma_{z_3} = (Q_{\boldsymbol{u}_3} \otimes Q_{\boldsymbol{u}_4} + \sigma_\varepsilon^{-2} A_3^{\mathrm{T}} A_3)^{-1}$. These posterior moment computations need the inversion of an $M_1 M_2 \times M_1 M_2$ matrix and cannot exploit the Kronecker structure because of the second summand in

Σ_{z_3}. To overcome this, we utilise the efficient method of Gilboa et al. (2015, Section 2.2), based on eigendecompositions and matrix-vector multiplications for Kronecker matrices. This procedure applies to the case when $A_3^{\mathrm{T}} A_3 = I_{M_1 M_2}$; this constraint requires the data to be observed on the complete grid (not necessarily equidistant), but can easily be relaxed for incomplete grids and domain extensions with an additional Gibbs step to sample the missing observations. Specifically, we make use of the identity

$$\Sigma_{z_3} = \left(Q_{\boldsymbol{u}_3} \otimes Q_{\boldsymbol{u}_4} + \sigma_\varepsilon^{-2} I_{M_1 M_2}\right)^{-1} = E_3 \otimes E_4 (\Lambda_3 \otimes \Lambda_4 + \sigma_\varepsilon^{-2} I_{M_1 M_2})^{-1} E_3^{\mathrm{T}} \otimes E_4^{\mathrm{T}}, \tag{4.10}$$

where $Q_{\boldsymbol{u}_3} = E_3 \Lambda_3 E_3^{\mathrm{T}}$ and $Q_{\boldsymbol{u}_4} = E_4 \Lambda_4 E_4^{\mathrm{T}}$, with E_3 and E_4 denoting the eigenvectors matrices and Λ_3 and Λ_4 denoting the diagonal matrices of eigenvalues of $Q_{\boldsymbol{u}_3}$ and $Q_{\boldsymbol{u}_4}$, respectively. The second crucial identity is

$$(E_3 \otimes E_4)\boldsymbol{\alpha} = \mathrm{vec}[(E_3[E_4 \,\mathrm{reshape}(\boldsymbol{\alpha}, M_2, M_1)]^{\mathrm{T}})^{\mathrm{T}}], \tag{4.11}$$

where the operator $\mathrm{reshape}(b, p, q)$ returns a $p \times q$ matrix whose elements are taken from the vector b, and $\mathrm{vec}(K)$ denotes the vectorisation of a matrix K.

Thus, to efficiently compute the posterior mean, $\boldsymbol{\mu}_{z_3}$, we follow three steps:

$$\begin{aligned}
\boldsymbol{\alpha} &= \mathrm{vec}\left[\left(E_3^{\mathrm{T}}[E_4^{\mathrm{T}} \,\mathrm{reshape}(\widetilde{\mathbf{y}}, M_2, M_1)]^{\mathrm{T}}\right)^{\mathrm{T}}\right], \\
\boldsymbol{\alpha} &= (\Lambda_3 \otimes \Lambda_4 + \sigma_\varepsilon^{-2} I_{M_1 M_2})^{-1} \boldsymbol{\alpha}, \\
\boldsymbol{\mu}_{z_3} &= \sigma_\varepsilon^{-2} \,\mathrm{vec}\left[\left(E_3[E_4 \,\mathrm{reshape}(\boldsymbol{\alpha}, M_2, M_1)]^{\mathrm{T}}\right)^{\mathrm{T}}\right],
\end{aligned}$$

where $\widetilde{\boldsymbol{y}} := \boldsymbol{y} - A_1 \boldsymbol{z}_1 - A_2 \boldsymbol{z}_2$. Note that $(\Lambda_3 \otimes \Lambda_4 + \sigma_\varepsilon^{-2} I_{M_1 M_2})$ is diagonal and therefore easy to invert. A posterior sample of $\boldsymbol{z}_3$ is then obtained by sampling $\boldsymbol{\eta} \sim \mathrm{N}(0, I_{M_1 M_2})$ and setting $\boldsymbol{z}_3 = \boldsymbol{\mu}_{z_3} + E_3 \otimes E_4 (\Lambda_3 \otimes \Lambda_4 + \sigma_\varepsilon^{-2} I_{M_1 M_2})^{-1/2} \boldsymbol{\eta}$, where for the latter operation, we again make use of the second identity in Eq. (4.11) and the diagonal form of $(\Lambda_3 \otimes \Lambda_4 + \sigma_\varepsilon^{-2} I_{M_1 M_2})$.

The last critical computation is the evaluation of the block marginal likelihood $\mathrm{N}(\widetilde{\boldsymbol{y}} \mid 0, Q_{\boldsymbol{u}_3}^{-1} \otimes Q_{\boldsymbol{u}_4}^{-1} + \sigma_\varepsilon^2 I_{M_1 M_2})$, which is required to sample $(\boldsymbol{\zeta}_3, \boldsymbol{\zeta}_4)$ and the corresponding hyperparameters, λ_3 and λ_4. First, the quadratic term can be calculated efficiently following the approach employed for the posterior mean. Next, for the log

determinant computation, one can use again the eigendecomposition; namely,

$$\log\det\left(Q_{\boldsymbol{u}_3}^{-1}\otimes Q_{\boldsymbol{u}_4}^{-1}+\sigma_\varepsilon^2 I_{M_1M_2}\right)^{-1}= \\ \log\det\left(E_3\otimes E_4(\Lambda_3^{-1}\otimes\Lambda_4^{-1}+\sigma_\varepsilon^2 I_{M_1M_2})^{-1}E_3^{\mathrm{T}}\otimes E_4^{\mathrm{T}}\right) \\ =-\log\det\left(\Lambda_3^{-1}\otimes\Lambda_4^{-1}+\sigma_\varepsilon^2 I_{M_1M_2}\right),$$

where $\Lambda_3^{-1}\otimes\Lambda_4^{-1}+\sigma_\varepsilon^2 I_{M_1M_2}$ is a diagonal matrix, whose log determinant is straightforward to calculate. We emphasize the required terms can also be efficiently computed for higher-order interactions through D-dimensional versions of the two key identities in Eq. (4.10) and Eq. (4.11) in Gilboa et al. (2015).

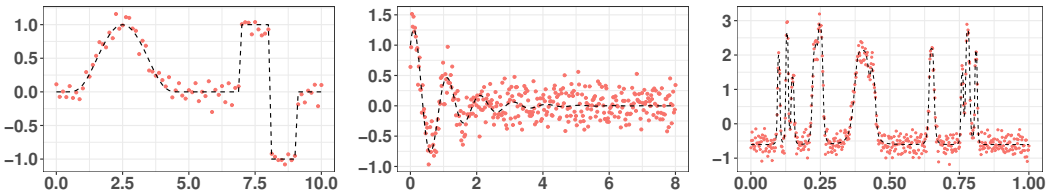
4.6 Experiments

We apply the sparse non-stationary hierarchical methodology to three simulated one-dimensional interpolation experiments and a two-dimensional synthetic example. First, the one-dimensional experiments study the effects of the discretisation and sample size on the efficiency of the algorithms presented in Section 4.4 under two extreme hyperpriors. In addition, the experiments show that our model can recover different signal types, while also providing information on the correlation structure. Second, a two-dimensional synthetic experiment demonstrates how the model can be extended to higher dimensions utilising an AGP model. Finally, in Section 4.6.3, we present a comparative evaluation on the performance of sparse 2-level GP models against two other methods: a stationary GP model and a Bayesian treed GP (TGP, Gramacy, 2007) model, a popular approach for dealing with non-stationarity.

4.6.1 One-dimensional synthetic data

We consider three simulated datasets with different signal types illustrated in Figure 4.4. The first example is a function with smooth parts and edges and is also piecewise constant. The second synthetic dataset is a damped sine wave function with smooth decaying oscillations. The third example corresponds to the *Bumps* function employed by Donoho and Johnstone (1995), which depicts a signal with pronounced spikes and constant parts.

In the first dataset, we investigate, empirically, posterior consistency of the estimates with respect to the discretisation scheme. The second experiment explores the performance of the sampling schemes for increased sample size and measurement noise. The last example emphasises the importance of the prior choice.



(a) Experiment 1 (b) Experiment 2 (c) Experiment 3

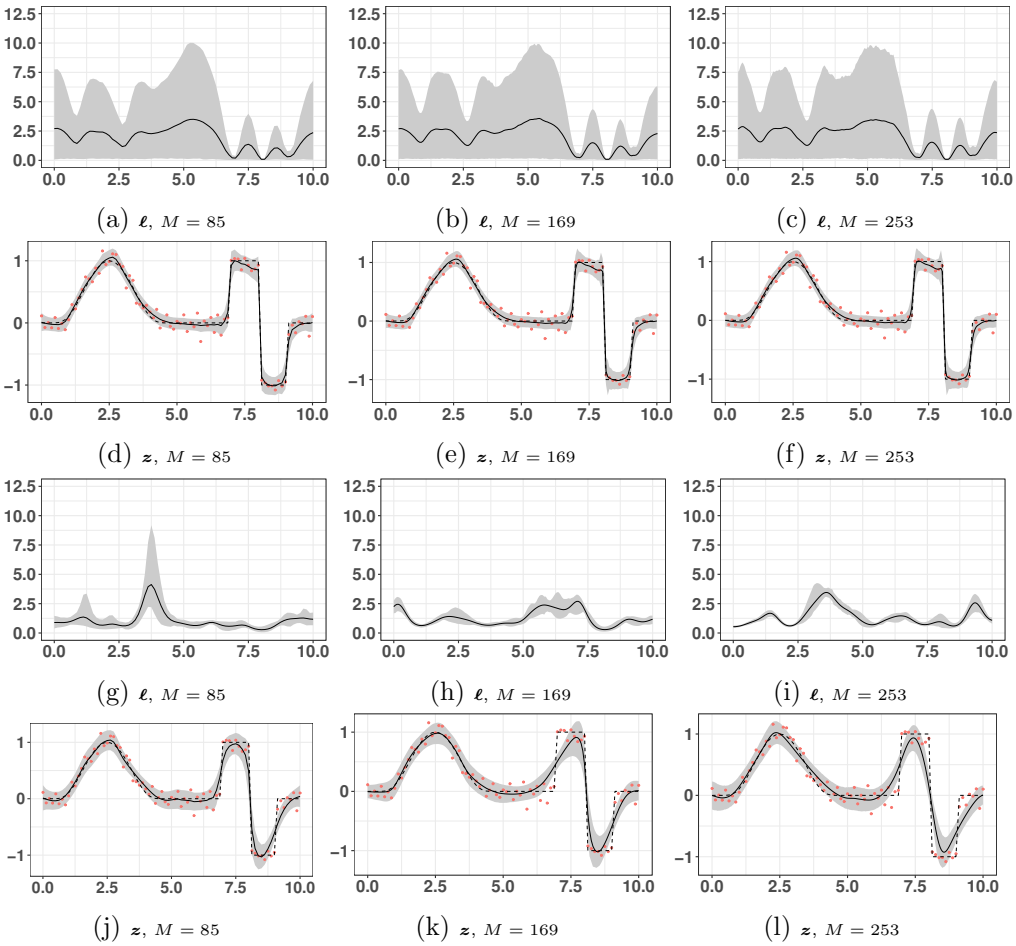
Figure 4.4: One-dimensional simulated dataset. (a): 81 observations with domain $[0, 10]$ and noise variance $\sigma_\varepsilon^2 = 0.01$. (b): 350 observations with domain $[0, 8]$ and noise variance $\sigma_\varepsilon^2 = 0.04$. (c): 512 observations with domain $[0, 1]$ and noise variance $\sigma_\varepsilon^2 = 0.04$.

4.6.1.1 Experiment 1: Smooth-piecewise constant function

For all experiments, we use the same initialisation and run the chains for $T = 200,000$ iterations. The burn-in period is algorithm specific, selected according to preliminary runs based on Raftery and Lewis's diagnostic (Raftery and Lewis, 1992) for the second level length-scale. Numerical discretisation-invariance is studied by varying M in the experiments, with $M = 85, 169$, and 253. The mean and variance of the prior length-scale process is set at zero and one, respectively. For the second level length-scale, we use a broad prior, $\log \lambda \sim \mathrm{N}(0, 3)$.

We start by presenting the results obtained with the MWG algorithm. Figure 4.5 shows estimates of the spatially varying length-scales and the unknown function under both hyperpriors. For the AR(1) hyperprior, an inspection of traceplots and cumulative averages of the estimates (not shown) suggest convergence of the chains for all discretisation schemes. In addition, the varying length-scale estimates exhibit the expected behaviour (i.e. decaying when the function has a sharp jump and increasing when the function is constant), and the interpolated estimates indicate a reasonable fit to the unknown function for all three discretisations schemes (Figure 4.5(a)-(f)). However, this is not the case for the SE hyperprior. Figure 4.5(g)-(l) illustrates the results obtained with this hyperprior for the same sampling algorithm. Under this setting, the effect of discretisation scheme is evident. As we increase M, the method fails to recover the unknown function. The strong correlation between the elements of $\boldsymbol{u}$ induced by the SE hyperprior makes the algorithm converge rather slowly to the target distribution.

In contrast to the results obtained with MWG, both w-Ell-SS and m-Ell-SS demonstrate convergence for both hyperpriors and invariance to the discretisation (see Figures C.1 and C.2 in the Appendix for a complete analysis). Figure 4.6 sum-



(a) $\boldsymbol{\ell}$, $M = 85$ (b) $\boldsymbol{\ell}$, $M = 169$ (c) $\boldsymbol{\ell}$, $M = 253$

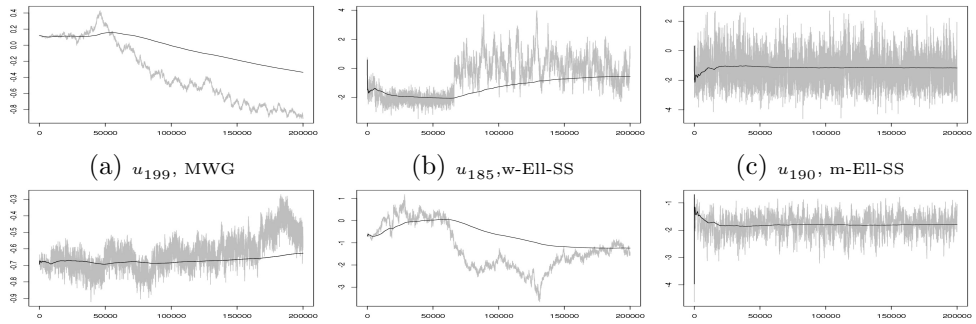
(d) $\boldsymbol{z}$, $M = 85$ (e) $\boldsymbol{z}$, $M = 169$ (f) $\boldsymbol{z}$, $M = 253$

(g) $\boldsymbol{\ell}$, $M = 85$ (h) $\boldsymbol{\ell}$, $M = 169$ (i) $\boldsymbol{\ell}$, $M = 253$

(j) $\boldsymbol{z}$, $M = 85$ (k) $\boldsymbol{z}$, $M = 169$ (l) $\boldsymbol{z}$, $M = 253$

Figure 4.5: Results for Experiment 1 with MWG. (a)-(c): Estimated $\boldsymbol{\ell}$ process with 95% credible intervals for AR(1) hyperprior on different grids. (d)-(f): Estimated $\boldsymbol{z}$ process with 95% credible intervals for AR(1) hyperprior on different grids with observed data in red. (g)-(i): Estimated $\boldsymbol{\ell}$ process with 95% credible intervals for SE hyperprior on different grids. (j)-(l): Estimated $\boldsymbol{z}$ process with 95% credible intervals for SE hyperprior on different grids with observed data in red.

marises succinctly important differences in mixing across the algorithms by showing traceplots with cumulative averages for a subset of parameters. The results are shown for the most challenging scenario, SE hyperprior at the highest resolution, $M = 253$. Figure 4.6(a)(d) emphasises the lack of convergence for MWG. Figure 4.6(b)(e) demonstrates the high autocorrelation of the chains and the slow convergence produced by w-Ell-SS. Finally, Figure 4.6(c)(f) highlights the improvement offered by m-Ell-SS, fast convergence to the stationary distribution and low autocorrelation of the chains. In order to evaluate the performance of the algorithms,



(a) u_{199}, MWG (b) u_{185},w-Ell-SS (c) u_{190}, m-Ell-SS

(d) λ, MWG (e) λ, w-Ell-SS (f) λ, m-Ell-SS

Figure 4.6: Traceplots with cumulative averages of the chains for SE hyperprior with $M = 253$. (Top row:) element of $\boldsymbol{u}$ with the lowest ESS. (Bottom row:) the hyperparameter.

we show in Table 4.1 an overall efficiency score (OES) of the chains (Titsias and Papaspiliopoulos, 2018). This measure considers both the CPU time (Table C.2 in the Appendix) required to run the chains and the effective sample size (ESS) (Table C.3 in the Appendix). The score is computed as OES = ESS/CPUtime[†]. For both multi-dimensional vectors, $\boldsymbol{z}$ and $\boldsymbol{u}$, we report the OES computed with the minimum ESS across all dimensions. The results indicate that while MWG with the AR(1) hyperprior shows high efficiency for some parameters when $M = 85$, its performance deteriorates as M increases. This suggests that this sampling scheme will not perform efficiently for bigger datasets even when $N = M$ (this is explored in Experiment 2). Furthermore, despite the fact that MWG reports the lowest CPU time under the AR(1) hyperprior (Table C.2 in the Appendix), its overall efficiency scores are outperformed by those obtained with m-Ell-SS; this is due to the low autocorrelation of the chains achieved by the marginal sampler (see Table C.3 in the Appendix). In contrast, chains of the parameters for w-Ell-SS result in the worse OES. Notice also that the scores reported for MWG with the SE hyperprior are not informative as the chains show convergence problems. Table 4.1 also reports mean absolute error (MAE) to evaluate the fit to the unknown function and the empirical coverage of the 95% credible intervals (EC) to evaluate accuracy in uncertainty quantification. For the SE hyperprior, w-Ell-SS and m-Ell-SS report equivalent errors and EC, while MWG yields worse values.

[†] All experiments were run in an Intel Core i7-6700 CPU (3.40GHz, 16 GB of RAM).

		MWG			w-Ell-SS			m-Ell-SS		
		$M = 85$	$M = 169$	$M = 253$	$M = 85$	$M = 169$	$M = 253$	$M = 85$	$M = 169$	$M = 253$
AR(1)	σ_ε^2	622.76	173.12	65.99	380.89	102.38	38.91	**661.20**	**257.81**	**116.35**
	ℓ_{min}	**635.36**	114.02	41.05	30.90	8.99	2.94	287.16	**114.36**	**59.71**
	z_{min}	**203.80**	42.10	13.91	9.12	2.34	0.86	129.75	**52.16**	**22.30**
	λ	89.84	15.66	6.00	22.77	5.26	2.36	**111.80**	**45.54**	**21.53**
	MAE	0.041	0.051	0.054	0.041	0.051	0.054	0.041	0.051	0.053
	EC	0.988	0.975	0.971	0.988	0.975	0.975	0.988	0.975	0.975
SE	σ_ε^2	11.19	4.88	7.49	246.24	77.72	8.89	**856.15**	**253.91**	**125.97**
	ℓ_{min}	1.22	0.73	0.64	21.69	10.22	2.79	**244.91**	**122.57**	**55.82**
	z	0.06	0.01	0.01	4.71	1.37	0.24	**76.80**	**24.11**	**9.87**
	λ	0.59	0.75	0.31	2.31	0.29	0.01	**16.59**	**4.15**	**2.21**
	MAE	0.078	0.100	0.133	0.040	0.050	0.054	0.039	0.049	0.052
	EC	0.889	0.826	0.763	0.988	0.975	0.971	0.988	0.975	0.979

Table 4.1: Experiment 1: OES with both hyperpriors under various discretisation schemes ($M = 85, 169, 253$) and three different algorithms. ℓ_{min} and z_{min} report OES for the minimum ESS across all dimensions. Highest values in boldface.

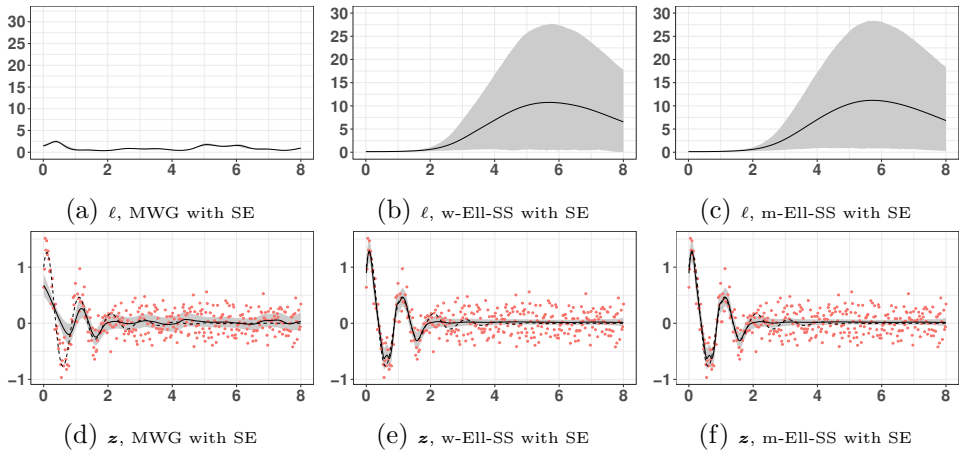
	AR(1)			SE		
	MWG	w-Ell-SS	m-Ell-SS	MWG	w-Ell-SS	m-Ell-SS
σ_ε^2	12.73	**27.54**	14.21	0.27	**32.29**	15.27
ℓ_{min}	0.06	0.14	**0.65**	0.00	0.40	**1.04**
z_{min}	0.13	0.13	**0.75**	0.01	0.55	**1.41**
λ	0.19	0.36	**0.95**	0.02	0.05	**0.25**
MAE	0.038	0.039	0.039	0.089	0.038	0.038
EC	0.920	0.934	0.934	0.863	0.940	0.934

Table 4.2: Experiment 2: OES with AR(1) and SE hyperprior employing three different algorithms. ℓ_{min} and z_{min} report OES for the minimum ESS across all dimensions. Highest values in boldface.

4.6.1.2 Experiment 2: Damped sine wave

This example explores the effect of increasing the sample size and measurement noise. Due to robustness of the estimates with respect to the discretisation in the first example, we only present experiments for the discretisation scheme when $N = M$. The chains are run for $T = 100,000$ iterations with a burn-in period that is algorithm and prior specific. In addition, we extend the domain with 40 points on each side of the interval, such that $M = 430$ and $N = 350$. The prior distributions for $\boldsymbol{u}$ and $\log \lambda$ are as in Experiment 1.

While the results with the AR(1) hyperprior appear satisfactory under the three sampling schemes (Figure C.3 in the Appendix), once again, SE hyperprior (Figure 4.7) with MWG is not able to explore the posterior of $\boldsymbol{u}$, resulting in poor estimates and hence, the highest MAE and poor EC (see Table 4.2). Analysing the



(a) ℓ, MWG with SE (b) ℓ, w-Ell-SS with SE (c) ℓ, m-Ell-SS with SE

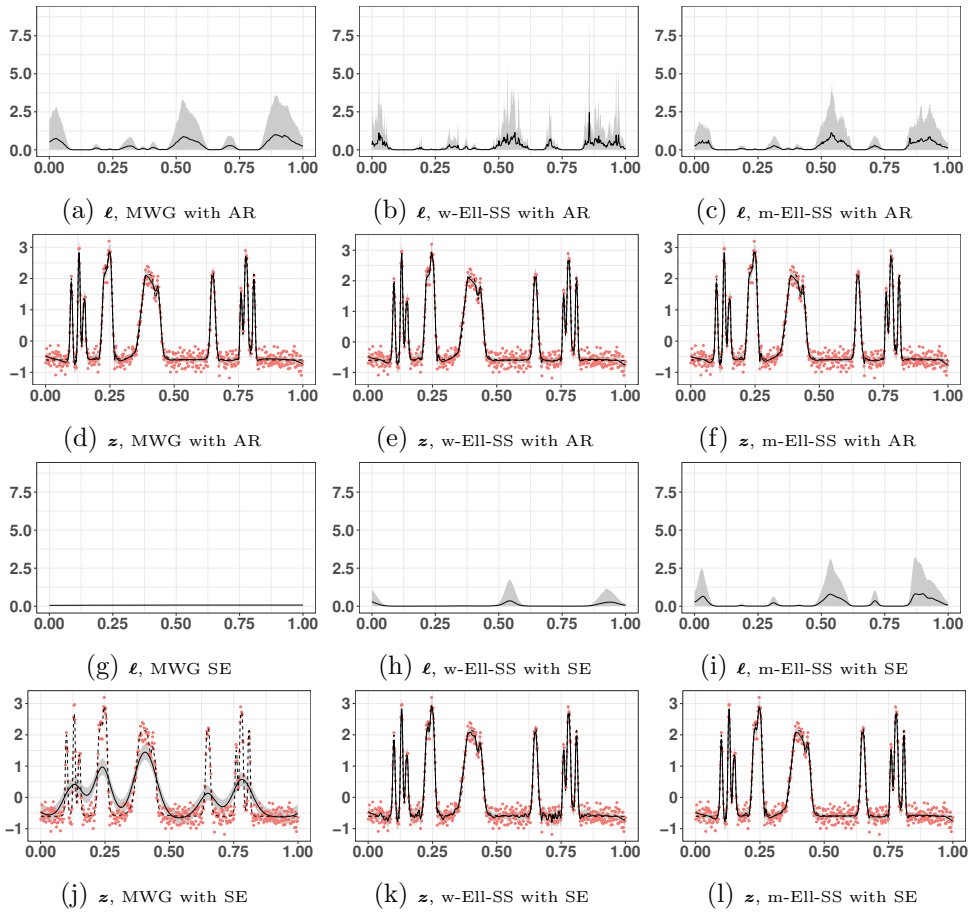
(d) $\boldsymbol{z}$, MWG with SE (e) $\boldsymbol{z}$, w-Ell-SS with SE (f) $\boldsymbol{z}$, m-Ell-SS with SE

Figure 4.7: Results for Experiment 2. Top row: estimated $\boldsymbol{\ell}$ process with 95% credible interval for SE hyperprior with (a) MWG, (b) w-Ell-SS and (c) m-Ell-SS. Second row: estimated $\boldsymbol{z}$ process with 95% credible interval for SE hyperprior with (d) MWG, (e) w-Ell-SS and (f) m-Ell-SS.

efficiency of the samplers, first, for the AR hyperprior, we observe that while MWG is faster (Table C.4 in the Appendix), its ESS is consistently smaller (Table C.6 in the Appendix), hence reducing its OES (Table 4.2). In contrast to the findings in Experiment 1, w-Ell-SS reports better OES compared to MWG due to better mixing in the chains. We believe this is due to the noise level, which favours a whitened parametrisation. Finally, despite the fact that the marginal sampler reports larger CPU times, the low correlation of its chains (Table C.6 in the Appendix) favours its OES. Second, when using the SE hyperprior, the marginal sampler appears to be significantly faster and consistently reports the best OES. This, together with the negligible differences in MAE and EC, suggests that m-Ell-SS offers a good compromise between computational cost and efficiency, with the benefit of working well under highly correlated priors.

4.6.1.3 Experiment 3: Bumps

The data is generated employing the *Bumps* function in Donoho and Johnstone (1995) and scaled to have zero mean and unit variance. Following Vannucci and Corradi (1999), we generate $N = 512$ points in the interval [0,1] and use a signal-to-noise ratio equal to 5, such that $\sigma_\varepsilon^2 = .04$. To avoid a boundary problem, we extend the domain with 30 points on each side of the interval, such that $M = 572$. Chains are run for $T = 100,000$ iterations with algorithm and prior specific burn-in periods. We use empirical priors for the log length-scale process and log length-scale



(a) $\boldsymbol{\ell}$, MWG with AR (b) $\boldsymbol{\ell}$, w-Ell-SS with AR (c) $\boldsymbol{\ell}$, m-Ell-SS with AR

(d) $\boldsymbol{z}$, MWG with AR (e) $\boldsymbol{z}$, w-Ell-SS with AR (f) $\boldsymbol{z}$, m-Ell-SS with AR

(g) $\boldsymbol{\ell}$, MWG SE (h) $\boldsymbol{\ell}$, w-Ell-SS with SE (i) $\boldsymbol{\ell}$, m-Ell-SS with SE

(j) $\boldsymbol{z}$, MWG with SE (k) $\boldsymbol{z}$, w-Ell-SS with SE (l) $\boldsymbol{z}$, m-Ell-SS with SE

Figure 4.8: Results for Experiment 3. Top row: estimated $\boldsymbol{\ell}$ process with 95% credible interval for AR(1) hyperprior with (a) MWG, (b) w-Ell-SS and (c) m-Ell-SS. Second row: estimated $\boldsymbol{z}$ process with 95% credible interval for AR(1) hyperprior with (d) MWG, (e) w-Ell-SS and (f) m-Ell-SS. Third row: estimated $\boldsymbol{\ell}$ process with 95% credible interval for SE hyperprior with (g) MWG, (h) w-Ell-SS and (i) m-Ell-SS. Bottom row: estimated $\boldsymbol{z}$ process with 95% credible interval for SE hyperprior with (j) MWG, (k) w-Ell-SS and (l) m-Ell-SS.

hyperparameter; namely, $\mu_u = -3.06$, $\tau_u^2 = 2.62$, and $\log \lambda \sim \mathrm{N}(-3.06, 2.62)$ (see Section C.1.3.1 in the Appendix for more details on prior elicitation).

This example highlights important differences between the two hyperpriors and the proposed MCMC algorithms. First, under the AR(1) hyperprior, the three sampling schemes show differences in the posterior length-scale process (Figure 4.8(a)-(c)). While MWG results in a smooth process, m-Ell-SS and w-Ell-SS appear to be more sensitive to the prior, with rougher estimates. Second, for the SE hyperprior, once more, MWG did not reach convergence. Also, the performance of w-Ell-SS

	AR(1)			SE		
	MWG	w-Ell-SS	m-Ell-SS	MWG	w-Ell-SS	m-Ell-SS
σ^2_ε	**23.42**	5.73	5.70	2.06	5.48	**15.36**
ℓ_{min}	0.01	0.01	**0.13**	0.00	0.01	**0.15**
z_{min}	**2.43**	0.10	0.24	0.56	0.07	**0.85**
λ	**0.65**	0.03	0.13	**0.07**	0.00	0.03
MAE	0.060	0.061	0.062	0.461	0.069	0.060
EC	0.955	0.950	0.959	0.385	0.961	0.967

Table 4.3: Experiment 3: OES with AR(1) and SE hyperprior employing three different algorithms. ℓ_{min} and z_{min} report OES for the minimum ESS across all dimensions. Highest values in boldface.

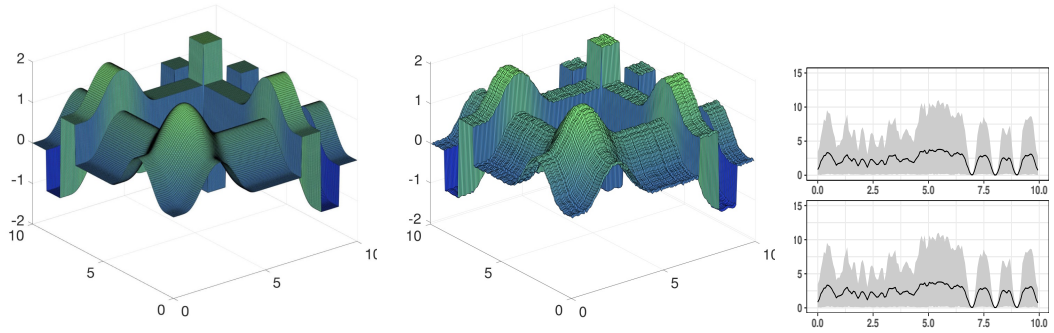
has become impaired; the posterior length-scale process does not reflect the changes in the correlation structure, and the length-scale hyperparameter did not reach the stationary distribution. The posterior length-scale process obtained with m-Ell-SS appears more appropriate, although, still shows a prior effect.

The findings discussed above are also evidenced in the OES shown in Table 4.3, where MWG exhibits the highest scores and the lowest MAE under AR(1). In contrast, the m-Ell-SS scheme outperforms MWG and w-Ell-SS for a SE hyperprior. We believe the differences illustrated in this experiment are a result of a key challenge of elliptical slice sampling. When the likelihood is strong, the sampler can result in poor mixing and, in extreme cases, can get stuck (Fagan et al., 2016). In addition, when sampling kernel parameters in strong likelihood settings, one can expect a centred parametrisation (avoiding whitening) to be more efficient (see Section 3 in Murray and Adams (2010)).

The computational time required for this experiment is reported in Table C.9 in the Appendix. Given the same initial values, the marginal sampler converges to the stationary distribution faster; indeed, m-Ell-SS reports, across experiments, the smallest time spent in burn-in period. Finally, to highlight how the model can benefit from using a more powerful computer, we ran this experiment in an Intel Xeon E5-260V3 2.4GHz (Haswell), 8-core processors with 4GB per core, and we found that the inference procedure is sped up by a factor of ≈ 2.1 for m-Ell-SS and w-Ell-SS (see Table C.10 in the Appendix). However, for MWG, the speed up factor was only ≈ 1.2.

4.6.2 Two-dimensional synthetic data

We study the performance of our approach on a 2-D synthetic dataset, by generating $N = 20,449$ noisy observations in an expanded grid of $M_1 = M_2 = 143$ equally



(a) True surface (b) Fitted surface and length-scale processes

Figure 4.9: Results for two-dimensional synthetic data. (a): True surface. (b): Posterior mean surface and one-dimensional length-scale processes with 95% credible intervals.

spaced points in $[0, 10]$, employing $z(x_1, x_2) = z(x_1) + z(x_2)$, where both $z(x_1)$ and $z(x_2)$ correspond to the function used in Experiment 1. The noise variance is set to $\sigma_\varepsilon^2 = .06$ and the sampler is run for $T = 50,000$ iterations, with a burn-in of $10,000$. We use the same prior distributions of Experiment 1 for each of the length-scale processes and corresponding hyperparameters.

Figure 4.9 depicts the true surface versus the posterior mean obtained from a 2-level AGP model (without interaction term), employing the block-m-Ell-SS algorithm. Our model is able to capture the smooth areas and edges of the surface. In addition, it provides information about the correlation structure along each axis (Figure 4.9(b)). The 2-level AGP correctly learns the varying correlation along the surface; for instance, the true function in the region $[5, 6] \times [5, 6]$ is constant, and in the same region, the 1-D length-scale processes depict strong correlation. The required total computational time for this experiment was 99.26 minutes (19.67 in burn-in and 79.59 in non-burned).

4.6.3 Comparative evaluation

We offer a comparative evaluation of our model for the synthetic examples from Section 4.6.1 and 4.6.2, against: 1) stationary Mátern Gaussian process (STAT) with $\nu = 1.5$ and 2) Bayesian treed Gaussian process (TGP). For the stationary model, the length scale and noise variance are inferred via MCMC, employing a marginal sampler with adaptive random walks. The GP prior mean and magnitude are fixed at 0 and 1, respectively, as in the sparse 2-level GP model. For the TGP, we consider a stationary Matern kernel with $\nu = 1.5$ and a constant mean function.

The magnitude is also inferred, in contrast to the stationary and the 2-level model. In order to make use of the default prior distributions, we rescale the response and inputs, as recommended by the authors.

In all the experiments, the chains are run for the same number of iterations $(100,000)$, with the same burnin period $(20,000)$, and initialised with the same values for STAT and sparse 2-level GP. For our two-dimensional simulated dataset (Experiment 4), we were unable to run the TGP model[†], due to the size of the dataset. To offer a comparison, we consider a subset of the original data, reducing the data size from $20,449$ to 441 observations.

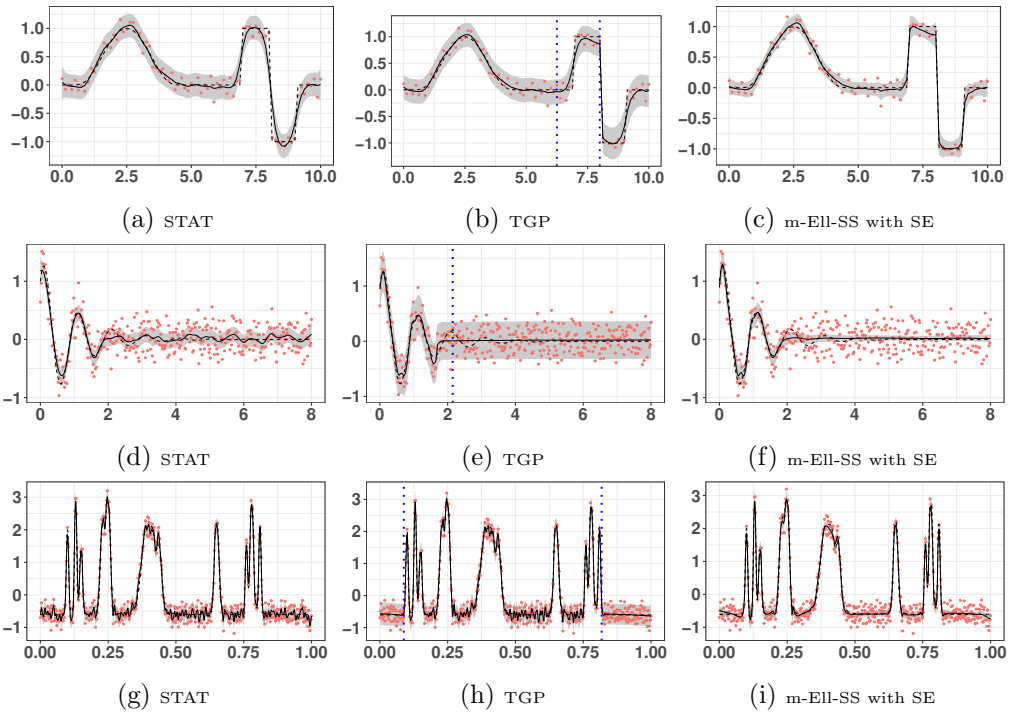
Figure 4.10 shows the posterior mean estimates of the unknown under the three models for the three different 1-D synthetic datasets, and Figure 4.11 illustrates the posterior mean surface for the subset of data in Experiment 4. In addition, Table 4.4 reports MAE and EC of the experiments. Note that the grey areas depict the 95% credible intervals of the unknown function for STAT and 2-level GP but, instead, depict the 95% credible intervals of the *noisy observations* for TGP. This is because storing region-specific traces is memory intensive, and the storage is not supported in the tgp package without doing predictions. Similarly, we report EC of the noisy process for TGP in Table 4.4.

		STAT		TGP		Sparse 2-level GP (AR/SE)	
	N	MAE	EC	MAE	EC*	MAE	EC
Experiment 1	81	0.076	0.914	0.056	0.963	0.041/**0.039**	0.988/0.988
Experiment 2	350	0.047	0.946	0.043	0.934	0.039/**0.038**	0.934/0.940
Experiment 3	512	0.094	0.947	0.079	0.963	0.062/**0.060**	0.959/0.967
Experiment 4 (subset)	441	0.195	0.501	0.122	0.980	**0.072**	0.963

Table 4.4: Comparative evaluation. For Experiments 1-3 with sparse 2-level GP model, we employ m-Ell-SS algorithm for both hyperpriors. Experiment 4 uses block-m-Ell-SS with AR hyperprior. EC* for TGP is reported for the noisy process. Best values in boldface.

First, the results make clear the downside of applying a stationary model to non-stationary data in all four experiments. In Experiment 1, STAT is oversmoothing and unable to capture the edges in the function (see Figure 4.10(a)). Example 2 and 3 (Figures 4.10(d) and (g)) illustrate how a stationary model tends to overfit when the function is constant, as a result of the different characteristics of the unknown. The same behaviour is repeated in the two-dimensional synthetic example

[†] A single iteration of TGP took more than 24 hours on an Intel Core i7-6700 CPU (3.40GHz, 16 GB of RAM). Also, we used TGP in an iMac Pro (2.3GHz 18-core Intel Xeon W processor, Turbo Boost up to 4.3GHz, 128GB 2666MHz DDR4 ECC memory) and after 2 weeks, the code was still running.

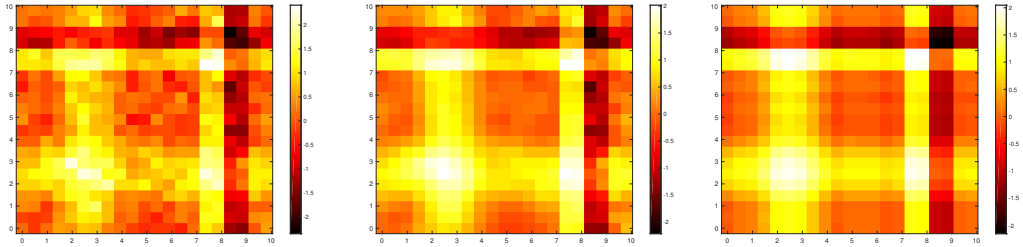


(a) STAT (b) TGP (c) m-Ell-SS with SE

(d) STAT (e) TGP (f) m-Ell-SS with SE

(g) STAT (h) TGP (i) m-Ell-SS with SE

Figure 4.10: Comparative evaluation for 1-D experiments. Each row shows one of the simulated experiments. Red dots depict observed data, dotted lines show the true signal, solid lines show the posterior mean, and grey areas depict 95% credible intervals. (a)(d)(g)(j): Stationary GP (b)(e)(h)(k): TGP, with blue dotted lines depicting MAP cut-off points. (c)(f)(i)(l): 2-level GP with m-Ell-SS algorithm and the hyperprior with lowest MAE. Grey area depict 95% credible intervals *of noisy observations* for TGP, while for STAT and 2-level they depict 95% credible intervals of the unknown function.



(a) STAT (b) TGP (c) Sparse 2-level GP

Figure 4.11: Comparative evaluation for 2-D experiment. Posterior mean surface for (a): anisotropic stationary model, (b): TGP, (c): 2-level AGP with first order terms.

(Figure 4.11(a)).

Second, while TGP offers an improvement, compared with a stationary setting, the model still oversmooths where the function possesses an edge. For instance, in Figures 4.10(b), the partition found around 6.2 is misplaced, and a third partition should be included around 9 to capture correctly the edges. In Experiment 2 (Figure 4.10(e)), the partition is also misplaced; this is however more reasonable (compared to Experiment 1) due to the smooth change in the behaviour. In Experiment 3, despite the fact that TGP fit is good when the function is constant (Figure 4.10(h)), the main limitation appears to be in finding some of the partitions that are required to ameliorate the issues resulting from fitting piecewise stationary models. Note that we ran TGP with a different number of iterations ($100,000$; $200,000$ and $500,000$) to verify the results shown in Figure 4.10 and 4.11 (see Section C.2 in the Appendix for the results). In Experiment 3, while increasing the number of iterations has a positive effect on the partitions found (and therefore on MAE), it was not enough to outperform the sparse 2-level GP model. Also, this was not the case for the other experiments, where increasing the number of iterations either did not affect the fit or worsened it. Moreover, without knowing the ground truth, it would be hard to know beforehand if the algorithm has been run for long enough to find the appropriate partitions.

In summary, the sparse 2-level GP is an alternative model for non-stationary data that resolves the issues discussed above. It does not overfit or oversmooth and appears to be more efficient in dealing with different types of non-stationarities, such as, edges, smooth changes, and sharp peaks. Moreover, the sparse 2-level GP clearly benefits from the additive structure, making the model scalable, while retaining flexibility. Notice that evaluating the methods solely on running time can be misleading, as STAT and 2-level GP are implemented in R using standard libraries, while TGP uses R as front end to call `C` and `C++` optimised code.

4.6.4 Real data: NASA rocket booster vehicle

The analysed dataset in this experiment comes from a computer simulator of a NASA rocket booster vehicle, the Langley Glide-Back Booster (Gramacy and Lee, 2012). NASA scientists are interested in understanding the behaviour of the rocket when it re-enters the atmosphere. To do so, the computer experiment considers six different output variables: lift, drag, pitch, side force, yaw, and roll; all forces that keep the rocket up, and three input variables: the speed (mach), the angle of attack (alpha), and the slide-slip angle (beta). Here, we focus on how the lift

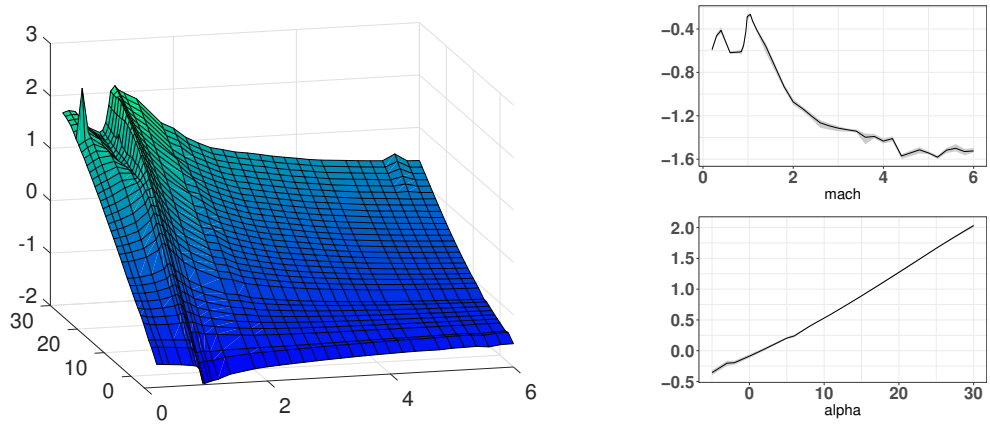


Figure 4.12: Results for NASA rocket booster vehicle. (Left:) Posterior mean. (Right:) Posterior mean of the two one-dimensional processes with 95% credible intervals.

force is affected as a function of the speed (mach) and the angle of attack (alpha) for a particular value of the slide-slip angle (beta=0). The data is, by nature, non-stationary, with different levels of smoothness along the surface and with a ridge showing the change from subsonic to supersonic flow at mach=1 and large alpha.

The data consists on 861 observations on a 34×33 grid where the speed ranges from $[.2, 6]$ and the angle of attack from $[-5, 30]$. The data is more dense for mach values around one. Thus, the data is available on an incomplete, non-equally spaced, rectangular grid. We consider the sparse 2-level AGP model with interaction term, employing the block-m-Ell-SS algorithm for inference. In order to deal with missing values, we use the model to impute them at each iteration of the MCMC. The chain is run for $50,000$ iterations with a burn-in period of $10,000$.

Figure 4.12 shows the posterior mean obtained. The model is able to capture the expected ridge around mach=1 and a sharp peak in the boundary around alpha=25, where the latter seems to be an error in the convergence of the simulator (Gramacy and Lee, 2012). Furthermore, the figure illustrates the posterior mean of each of the one-dimensional processes. The results suggest that fitting a stationary process for the angle of attack (alpha) may be enough. A depiction of the posterior mean of the second-order interaction term is provided in Figure 4.14, and visualisations of the posterior of all length scale processes are provided in Figure 4.13. The required computational time for this experiment was 5.78 hours in a high performance cluster.

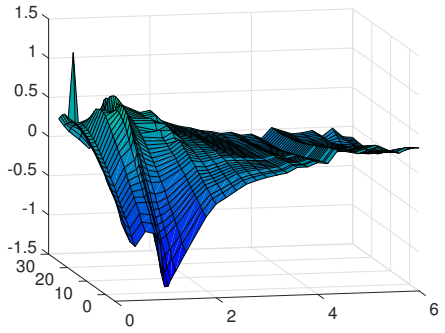
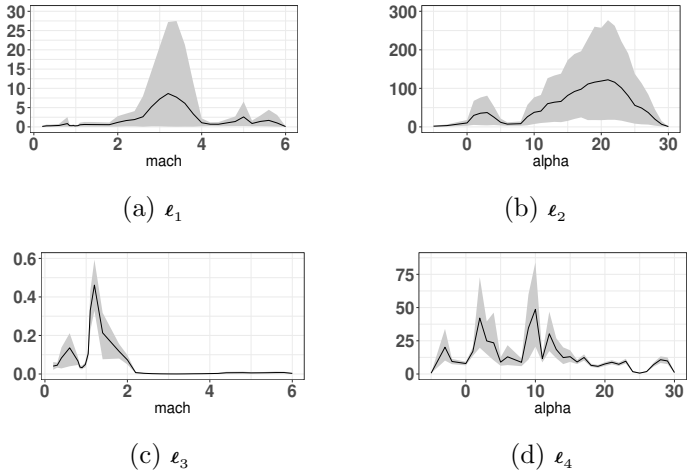


Figure 4.13: Results for NASA rocket booster vehicle experiment. Posterior mean of non-stationary interaction term.



(a) ℓ_1 (b) ℓ_2

(c) ℓ_3 (d) ℓ_4

Figure 4.14: Results for NASA rocket booster vehicle experiment. Posterior mean estimates of the stationary, one-dimensional length-scale processes with 95% credible intervals. (a): Length-scale process for $\boldsymbol{z}_1$. (b): Length-scale process for $\boldsymbol{z}_2$. (c)-(d): Length-scale processes for the interaction term, $\boldsymbol{z}_3$. Notice a dip of ℓ_4 at alpha=25 to recover the peak and the small values of ℓ_3 around mach=1.

4.7 Discussion

In this chapter we introduced novel non-stationary hierarchical models based on stochastic parameters and Gaussian Markov random fields, ameliorating the computational constraints of doing exact inference in 2-level GP models through sparsity in the finite-dimensional approximation of the inverse covariance matrix of the non-stationary field. Different hyperpriors were also explored for the spatially varying

length-scale, from strong prior smoothness assumptions through a squared exponential covariance to rough hyperpriors of an autoregressive AR(1) model, with the latter benefiting from further computational gains. Strong dependence between the model layers makes efficient inference challenging, and to address this, we introduced and investigated the performance of three different MCMC algorithms. First, we found that the Metropolis-within-Gibbs scheme performs poorly for highly correlated hyperpriors and exhibits deteriorating efficiency as the number of observations or discretisation size increase. Second, the whitened elliptical slice sampler performs well for weak likelihoods, regardless the hyperprior employed, at the price of highly correlated chains. Finally, the marginal elliptical slice sampler appears to be an efficient strategy to break the correlation between latent process and hyperparameters and offers a good compromise between computational complexity and efficiency of the chains.

We also proposed a novel extension to D-dimensional settings by combining additive Gaussian process models with sparse 2-level GPs. The additive structure and use of Kronecker algebra for the interaction term result in an inference procedure that is tractable and scalable. Our experiments show that the additive structure retains the flexibility of the 2-level GP and favours its interpretability. Moreover, while we focus on the two-dimensional setting, the additive 2-level model and inference scheme naturally extend to higher dimensions. Overall, the comparative evaluation highlights the benefits of our approach over stationary and popular non-stationary GP models, to recover edges, peaks and smooth variations in the data in both one-dimensional and two-dimensional settings. In addition, the methodology may benefit greatly from using powerful computational resources.

The experiments presented here suggest that the algorithms based on elliptical slice sampling do not deteriorate as the resolution becomes finer or the sample size increases, similar to the schemes discussed by Chen et al. (2019). However, it is important to emphasise that elliptical slice sampling is known to perform well for weak data likelihoods; therefore, care must be taken in the small noise limit. Furthermore, it would be interesting to explore the performance of the auxiliary gradient-based sampling scheme recently proposed by Titsias and Papaspiliopoulos (2018); however, notice that this scheme requires derivatives, which for our model are expensive and not straightforward to compute. We also highlight the recent work of Durrande et al. (2019), implementing banded matrix operators in TensorFlow, which, combined with GPflow (Matthews et al., 2017), could provide a promising direction for automatic differentiation for our model.

A natural extension of this work is to the 3-level GP model or, more generally,

the deep GP models studied in Dunlop et al. (2018). Other interesting directions for future research include exploring higher-order autoregressive hyperpriors; more general kernels; and alternative likelihoods for problems beyond regression, such as the classification and inverse problems discussed in Chen et al. (2019).

Algorithm 8 Block marginal elliptical slice sampling (block-m-Ell-SS)

Require: A_1, A_2, A_3 $\sigma_\varepsilon^{2(0)}$, $\boldsymbol{z}_1^{(0)}$, $\boldsymbol{z}_2^{(0)}$, $\boldsymbol{z}_3^{(0)}$, $\boldsymbol{\xi}_1^{(0)}$, $\boldsymbol{\xi}_2^{(0)}$, $\boldsymbol{\xi}_3^{(0)}$, $\boldsymbol{\xi}_4^{(0)}$, $\lambda_1^{(0)}$, $\lambda_2^{(0)}$, $\lambda_3^{(0)}$ and $\lambda_4^{(0)}$

1: **for** $t = 1$ to T **do**

2: Draw: $\log \sigma_\varepsilon^{2(t)}$ using RW-MH($\log \sigma_\varepsilon^{2(t-1)}, s_1$) ▷Alg. 2

Step 3 implies computing: $\min\left\{1, \frac{\mathrm{N}\left(\boldsymbol{y}|A_1\boldsymbol{z}_1+A_2\boldsymbol{z}_2+A_3\boldsymbol{z}_3,\sigma_\varepsilon^2 I_N\right)\pi\left(\log\sigma_\varepsilon^2\right)}{\mathrm{N}\left(\boldsymbol{y}|A_1\boldsymbol{z}_1+A_2\boldsymbol{z}_2+A_3\boldsymbol{z}_3,\sigma_\varepsilon^{2(t-1)} I_N\right)\pi\left(\log\sigma_\varepsilon^{2(t-1)}\right)}\right\}$

3: Run Adaptation for s_1

4: Draw: $\boldsymbol{\zeta}_1^{(t)}$ using Ell-SS($\boldsymbol{\zeta}_1^{(t-1)}, I_{M_1}$) ▷Alg. 4

Evaluation of log likelihood in Step 11 implies:

a. Computing: $\boldsymbol{u}_1' = R_{\lambda_1{}^{(t-1)}}\boldsymbol{\zeta}_1' + \boldsymbol{\mu}_{u_1}$

b. Computing: $\log \mathrm{N}(\boldsymbol{y} - A_2\boldsymbol{z}_2^{(t-1)} - A_3\boldsymbol{z}_3^{(t-1)} \mid 0, A_1 Q_{\boldsymbol{u}_1'}^{-1} A_1^{\mathrm{T}} + \sigma_\varepsilon^{2(t)} I_N)$

5: Draw: $\log \lambda_1^{(t)}$ using RW-MH($\log \lambda_1^{(t-1)}, s_2$) ▷Alg. 2

Step 3 implies:

a. Computing: $R_{\lambda_1'}$

b. Updating: $\boldsymbol{u}_1' = R_{\lambda_1'}\boldsymbol{\zeta}_1^{(t)} + \boldsymbol{\mu}_{u_1}$

c. Computing: $\min\left\{1, \frac{\mathrm{N}\left(\boldsymbol{y}-A_2\boldsymbol{z}_2^{(t-1)}-A_3\boldsymbol{z}_3^{(t-1)}|0,A_1Q_{\boldsymbol{u}_1'}^{-1}A_1^{\mathrm{T}}+\sigma_\varepsilon^{2(t)}I_N\right)\pi\left(\log\lambda_1'\right)}{\mathrm{N}\left(\boldsymbol{y}-A_2\boldsymbol{z}_2^{(t-1)}-A_3\boldsymbol{z}_3^{(t-1)}|0,A_1Q_{\boldsymbol{u}_1}^{-1}A_1^{\mathrm{T}}+\sigma_\varepsilon^{2(t)}I_N\right)\pi\left(\log\lambda_1^{(t-1)}\right)}\right\}$

6: Run Adaptation for s_2

7: Draw $\boldsymbol{z}_1^{(t)} = \mathrm{N}\left(\sigma_\varepsilon^{-2(t)}\Sigma_{z_1}A_1^{\mathrm{T}}(\boldsymbol{y} - A_2\boldsymbol{z}_2^{(t-1)} - A_3\boldsymbol{z}_3^{(t-1)}), \Sigma_{z_1}\right)$ ▷$\Sigma_{z_1}=(Q_{\boldsymbol{u}_1^{(t)}}+\sigma_\varepsilon^{-2(t)}A_1^{\mathrm{T}}A_1)^{-1}$

8: Repeat steps 4-7 for $\boldsymbol{z}_2, \boldsymbol{\zeta}_2, \lambda_2$

9: Draw: $\boldsymbol{\zeta}_{3,4}^{(t)}$ using Ell-SS($\boldsymbol{\zeta}_1^{(t-1)}, I_{M_1M_2}$) ▷Alg. 4

Evaluation of log likelihood in Step 11 implies: ▷ $\boldsymbol{\zeta}_{3,4}$ is formed by stacking $\boldsymbol{\zeta}_3$ and $\boldsymbol{\zeta}_4$

a. Updating: $\boldsymbol{u}_3' = R_{\lambda_3^{(t-1)}}\boldsymbol{\zeta}_3' + \boldsymbol{\mu}_{u_3}$ and $\boldsymbol{u}_4' = R_{\lambda_4^{(t-1)}}\boldsymbol{\zeta}_4' + \boldsymbol{\mu}_{u_4}$

b. Computing: $\log \mathrm{N}(\boldsymbol{y} - A_1\boldsymbol{z}_1^{(t)} - A_2\boldsymbol{z}_2^{(t)} \mid 0, A_3(Q_{\boldsymbol{u}_3'}^{-1} \otimes Q_{\boldsymbol{u}_4'}^{-1})A_3^{\mathrm{T}} + \sigma_\varepsilon^{2(t)} I_N)$

10: Draw: $\log \lambda_3^{(t)}$ using RW-MH($\log \lambda_3^{(t-1)}, s_3$) ▷Alg. 2

Step 3 implies:

a. Computing: $R_{\lambda_3'}$

b. Updating: $\boldsymbol{u}_3' = R_{\lambda_3'}\boldsymbol{\zeta}_3^{(t)} + \boldsymbol{\mu}_{u_3}$

c. Computing: $\min\left\{1, \frac{\mathrm{N}\left(\boldsymbol{y}-A_1\boldsymbol{z}_1^{(t)}-A_2\boldsymbol{z}_2^{(t)}|0,A_3(Q_{\boldsymbol{u}_3'}^{-1}\otimes Q_{\boldsymbol{u}_4}^{-1})A_3^{\mathrm{T}}+\sigma_\varepsilon^{2(t)}I_N\right)\pi\left(\log\lambda_3'\right)}{\mathrm{N}\left(\boldsymbol{y}-A_1\boldsymbol{z}_1^{(t)}-A_2\boldsymbol{z}_2^{(t)}|0,A_3(Q_{\boldsymbol{u}_3}^{-1}\otimes Q_{\boldsymbol{u}_4}^{-1})A_3^{\mathrm{T}}+\sigma_\varepsilon^{2(t)}I_N\right)\pi\left(\log\lambda_3^{(t-1)}\right)}\right\}$

11: Run Adaptation for s_3

12: Repeat 10-11 for λ_4

13: Draw $\boldsymbol{z}_3^{(t)} = \mathrm{N}\left(\sigma_\varepsilon^{-2(t)}\Sigma_{z_3}A_3^{\mathrm{T}}(\boldsymbol{y} - A_1\boldsymbol{z}_1^{(t-1)} - A_2\boldsymbol{z}_2^{(t-1)}), \Sigma_{z_3}\right)$ ▷$\Sigma_{z_3}=(Q_{\boldsymbol{u}_3^{(t)}}\otimes Q_{\boldsymbol{u}_4^{(t)}}+\sigma_\varepsilon^{-2(t)}A_3^{\mathrm{T}}A_3)^{-1}$

14: **end for**

CHAPTER 5

A NON-STATIONARY VARIATIONALLY SPARSE MCMC

This chapter introduces a sparse variational Markov chain Monte Carlo (MCMC) method for multi-level Gaussian process models, extending the work of Hensman et al. (2015). The scheme presented here combines the computational advantages of variational inference over a sparse Gaussian process model with the flexibility and convergence guarantees of MCMC methods. This work employs the inducing point framework (Snelson and Ghahramani, 2006; Titsias, 2009) to derive an optimal low-dimensional variational posterior distribution which minimises the Kullback-Leibler (KL) divergence between the augmented approximated and the augmented true posterior distribution. The derived approximated posterior factorises across data points, reducing the complexity to scale linearly in the number of observations. However, for each observation, the required exponentiated expected log-likelihood is intractable. Following Hensman et al. (2015), we employ Gauss-Hermite quadrature to approximate the intractable expectations. Simulation studies indicate that our model necessitates high-order Gauss-Hermite quadrature approximations to explore the posterior efficiently. This undermines the computational benefits of the variational inducing framework, opening interesting directions of research discussed in detail in Chapter 6.

The work here presented is in collaboration with Dr Sara Wade.

5.1 Introduction

Variational methods are commonly employed in the machine learning literature as a faster, approximate alternative to MCMC schemes (Blei et al., 2017). More pre-

cisely, for Gaussian process models variational inference is an active area of research (e.g. Titsias, 2009; Matthews et al., 2016; Hensman et al., 2013; Cutajar et al., 2019; Damianou and Lawrence, 2013) to ameliorate the computational burden of working with GPs.

Hensman et al. (2015) combined the inducing point framework of Snelson and Ghahramani (2006) with variational inference and MCMC methods to derive a scalable yet flexible framework for Gaussian process (GP) models. The sparse variational method results in a low-dimensional approximate posterior, and MCMC can be employed to draw from this complicated posterior (this idea was initially pointed out by Titsias et al. (2011)). Thus, the variationally sparse MCMC framework benefits from (i) the sparse variational method to alleviate the computational burden and (ii) the flexibility and theoretical guarantees of MCMC to sample from the complicated low-dimensional variational posterior. Importantly, this last step avoids placing any further distributional assumptions on the low-dimensional posterior, e.g. independence, that are typically required in full variational schemes.

We begin with a description of the variationally sparse MCMC framework for a standard, single-level GP model introduced in Hensman et al. (2015). The observed data consists of outputs y_n, which may be real-valued or more generally binary, counts, etc., with corresponding input locations $\boldsymbol{x}_n \in \mathbb{R}^D$ for $n = 1, \ldots, N$. The likelihood is assumed to factorise across data points, dependent on an unknown function $z : \mathbb{R}^D \to \mathbb{R}$ that maps the input locations to the real line:

$$p(\boldsymbol{y} \mid \boldsymbol{z}, \boldsymbol{\rho}) = \prod_{n=1}^{N} p(y_n \mid z(\boldsymbol{x}_n), \boldsymbol{\rho}),$$

where $\boldsymbol{y} = (y_1, \ldots, y_N)^T$, $\boldsymbol{z} = (z_1, \ldots, z_N)^T$ with $z_n \equiv z(\boldsymbol{x}_n)$, and $\boldsymbol{\rho}$ contains any additional likelihood parameters. The unknown function z has a Gaussian process prior with zero mean and covariance function $C_{\boldsymbol{\phi}}(\cdot, \cdot)$ parametrised by $\boldsymbol{\phi}$, namely, $z(\cdot) \sim \mathrm{GP}(0, C_{\boldsymbol{\phi}}(\cdot, \cdot))$.

As discussed in Chapter 3, a huge burden to employ GPs in practice is the high computational complexity, which scales cubically with the number of data points. To overcome this, Snelson and Ghahramani (2006) proposed the sparse pseudo-input framework. The key idea of this approach is to augment the data with a set of $M \ll N$ inducing or pseudo-points $\widetilde{X} = (\widetilde{\boldsymbol{x}}_1, \ldots \widetilde{\boldsymbol{x}}_M)^T$ and collect the values of the latent functions at the inducing points into the vectors $\widetilde{\boldsymbol{z}} = (\widetilde{z}_1, \ldots, \widetilde{z}_M)^T$, where $\widetilde{\boldsymbol{z}}_m \equiv z(\widetilde{\boldsymbol{x}}_m)$; we refer to the $\widetilde{z}_m$ as the *inducing variables*. By the properties of GPs,

the conditional augmented distribution is

$$\begin{aligned}\pi(\boldsymbol{z}, \widetilde{\boldsymbol{z}} \mid \boldsymbol{\phi}) &= \pi(\boldsymbol{z} \mid \widetilde{\boldsymbol{z}}, \boldsymbol{\phi})\pi(\widetilde{\boldsymbol{z}} \mid \boldsymbol{\phi}) \\ &= \mathrm{N}\left(\boldsymbol{z} \mid C_{\boldsymbol{z},\widetilde{\boldsymbol{z}}} C_{\widetilde{\boldsymbol{z}},\widetilde{\boldsymbol{z}}}^{-1} \widetilde{\boldsymbol{z}}, C_{\boldsymbol{z},\boldsymbol{z}} - C_{\boldsymbol{z},\widetilde{\boldsymbol{z}}} C_{\widetilde{\boldsymbol{z}},\widetilde{\boldsymbol{z}}}^{-1} C_{\widetilde{\boldsymbol{z}},\boldsymbol{z}}\right) \mathrm{N}\left(\widetilde{\boldsymbol{z}} \mid 0, C_{\widetilde{\boldsymbol{z}},\widetilde{\boldsymbol{z}}}\right),\end{aligned}$$

where we make use of the short notation $C_{\boldsymbol{z},\boldsymbol{z}}$ and $C_{\widetilde{\boldsymbol{z}},\widetilde{\boldsymbol{z}}}$ to denote the covariance matrix constructed by evaluating the kernel at the inputs X and $\widetilde{X}$, respectively, and $C_{\boldsymbol{z},\widetilde{\boldsymbol{z}}}$ to denote the cross-covariance matrix between the the function evaluated at the inputs X and inducing points $\widetilde{X}$. Under the augmented model, the posterior of the parameters and latent variables is given by:

$$\pi(\boldsymbol{z}, \widetilde{\boldsymbol{z}}, \boldsymbol{\rho}, \boldsymbol{\phi} \mid \boldsymbol{y}, X) \propto \prod_{n=1}^{N} p(y_n \mid z_n, \boldsymbol{\rho})\pi(\boldsymbol{z} \mid \widetilde{\boldsymbol{z}}, \boldsymbol{\phi})\pi(\widetilde{\boldsymbol{z}} \mid \boldsymbol{\phi})\pi(\boldsymbol{\phi})\pi(\boldsymbol{\rho}).$$

The variationally sparse approach restricts the approximate variational posterior to take the form:

$$q(\boldsymbol{z}, \widetilde{\boldsymbol{z}}, \boldsymbol{\rho}, \boldsymbol{\phi}) \propto \pi(\boldsymbol{z}|\widetilde{\boldsymbol{z}}, \boldsymbol{\phi})q(\widetilde{\boldsymbol{z}}, \boldsymbol{\rho}, \boldsymbol{\phi}). \tag{5.1}$$

Note that in the right-hand side of Eq. (5.1), the first term corresponds to the prior predictive distribution of $\boldsymbol{z}$ given $\widetilde{\boldsymbol{z}}$, while the second is the joint approximated posterior of the inducing variables, parameters, and hyperparameters. Thus, the variationally sparse approach assumes that conditioned on the inducing variables and hyperparameters, the latent function at the observed input locations does not depend on the data. This assumption is crucial to achieve the desired scalability, but the accuracy of this approximation clearly depends on the number and locations of the inducing points. Under this assumption, Hensman et al. (2015) showed that the optimal variational posterior which minimizes the KL divergence between the approximate and true posterior, $\mathrm{KL}(q(\boldsymbol{z}, \widetilde{\boldsymbol{z}}, \boldsymbol{\rho}, \boldsymbol{\phi}) \parallel \pi(\boldsymbol{z}, \widetilde{\boldsymbol{z}}, \boldsymbol{\rho}, \boldsymbol{\phi} \mid \boldsymbol{y}, X))$, corresponds to Eq. (5.1), where the lower dimensional variational posterior (the second term in Eq. (5.1)) takes the form

$$q(\widetilde{\boldsymbol{z}}, \boldsymbol{\rho}, \boldsymbol{\phi}) \propto \exp\left(\sum_{n=1}^{N} \mathbb{E}_{\pi(z_n|\widetilde{\boldsymbol{z}},\boldsymbol{\phi})}[\log(p(y_n \mid z_n, \boldsymbol{\rho}))]\right) \pi(\widetilde{\boldsymbol{z}} \mid \boldsymbol{\phi})\pi(\boldsymbol{\phi})\pi(\boldsymbol{\rho}). \tag{5.2}$$

The computation of Eq. (5.2) involves expectations over univariate Gaussian random variables, and when not available analytically, Hensman et al. (2015) suggest to approximate them with Gauss-Hermite quadrature. In addition, to avoid placing any further restrictions on the form of the optimal variational posterior, Hensman

et al. (2015) propose to use MCMC methods to sample from Eq. (5.2).

This chapter builds upon the ideas described above to develop a variationally sparse MCMC method for multi-level GP models. This hybrid (VI/MCMC) strategy is well-suited to our model because the complexity of the model, in the number of parameters and high correlation among them, (i) makes standard mean-field variational methods, which make strong independence assumptions, unsuitable and (ii) makes MCMC inference over the true posterior computationally expensive and challenging (see Chapter 3).

This chapter is organised as follows. We start, in Section 5.2, with a review of the non-stationary kernel at the heart of our 2-level GP models. This section considers different formulations of the kernel matrix parameter in D-dimensional input settings, emphasizing how the different formulations result in different assumptions about the non-stationary covariance, and therefore in different posteriors. Section 5.3 derives the optimal variationally sparse posterior distributions for each of the formulations of the kernel matrices described. In addition, it includes detailed algorithms to sample from the distributions of interest. Finally, Section 5.4 presents simulation studies that highlight the computational advantages of the proposed scheme and also the importance of the quadrature approximation.

5.2 The kernel matrices

A 2-level non-stationary Gaussian process prior for $z(\cdot)$ can be constructed when employing the family of non-stationary covariance functions (Paciorek and Schervish, 2006),

$$C_{\boldsymbol{\phi}}^{\mathrm{NS}}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \tau_z^2 \frac{|\Sigma(\boldsymbol{x}_i)|^{\frac{1}{4}} |\Sigma(\boldsymbol{x}_j)|^{\frac{1}{4}}}{|(\Sigma(\boldsymbol{x}_i) + \Sigma(\boldsymbol{x}_j))/2|^{\frac{1}{2}}} R_\psi\left(G_{ij}\right), \tag{5.3}$$

with $G_{ij} = \sqrt{(\boldsymbol{x}_i - \boldsymbol{x}_j)^T \left((\Sigma(\boldsymbol{x}_i) + \Sigma(\boldsymbol{x}_j))/2\right)^{-1} (\boldsymbol{x}_i - \boldsymbol{x}_j)}$, R_ψ a stationary correlation function on $\mathbb{R}$, and $\Sigma(\cdot)$ a $D \times D$ spatially varying covariance matrix, referred as the kernel matrix. Thus, the parameters $\boldsymbol{\phi}$ in the non-stationary covariance function consist of the magnitude, τ_z^2; the spatially varying covariance matrices, $\Sigma(\cdot)$; and any additional parameters ψ of the stationary correlation function R_ψ.

Importantly, non-stationarity is introduced in Eq. (5.3) through the kernel matrices, and this parameter must be inferred at every location where the process is observed. In Chapter 2, we provide a review on several approaches to model this spatially varying parameter and introduce a novel parametrisation for $D > 1$ based on

LDL^T factorisation. The LDL^T framework results in a flexible and general construction by permitting the kernel matrices to control not only the range of dependence, but also the direction. Nonetheless, this parametrisation can be problematic for high-dimensional problems as the number of processes needed for a D-dimensional setting is $D + D(D-1)/2$. This chapter studies alternative formulations of the kernel matrix parameter by making assumptions about the non-stationary kernel with the objective of contolling the number of parameters to make the sparse variational MCMC scheme suitable.

5.2.1 ARD covariance

By assuming a diagonal structure of the kernel matrices, we obtain a non-stationary automatic relevance determination (ARD) kernel. This formulation allows the behaviour of the correlation to be different in each dimension, with the assumption that the processes describing that correlation structure are independent. More precisely, the kernel matrices are $\Sigma(\cdot) := \text{diag}(\ell_1^2(\cdot), \ldots, \ell_D^2(\cdot))$, where $\ell_d(\cdot)$ denote the independent spatially varying length-scale processes, for dimension $d = 1, \ldots, D$. Additionally, we assume that each $\ell_d(\cdot)$ is a function of the d^{th} dimension of the input only, resulting in a more parsimonious model with only (D) hyperparameters of the spatially varying length-scale, in contrast to the (D^2) hyperparameters in the general case. In this setting, the non-stationary covariance function in Eq. (5.3) simplifies to

$$C_{\boldsymbol{\phi}}^{\text{NS}}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \tau_z^2 \prod_{d=1}^{D} \frac{\ell_d(x_{id})^{\frac{1}{2}} \ell_d(x_{jd})^{\frac{1}{2}}}{\left([\ell_d^2(x_{id}) + \ell_d^2(x_{jd})]/2\right)^{\frac{1}{2}}} R_\psi \left(\sqrt{\sum_{d=1}^{D} \frac{2(x_{id} - x_{jd})^2}{\ell_d^2(x_{id}) + \ell_d^2(x_{jd})}} \right). \tag{5.4}$$

Figure 5.1 illustrates a realisation of a Matérn process with a non-stationary ARD covariance function, with corresponding kernel matrices shown in Figure 5.4(a). Note that the diagonal structure in the kernel matrices results in axis alligned ellipses.

5.2.2 Isotropic covariance

The non-stationary isotropic covariance function is obtained by assuming the kernel matrices are scaled identity matrices such that $\Sigma(\cdot) = \ell^2(\cdot) I_D$. In this case, we can allow the spatially varying length-scale to be a function of the full D-dimensional input, with $\mathcal{O}(D)$ hyperparameters.

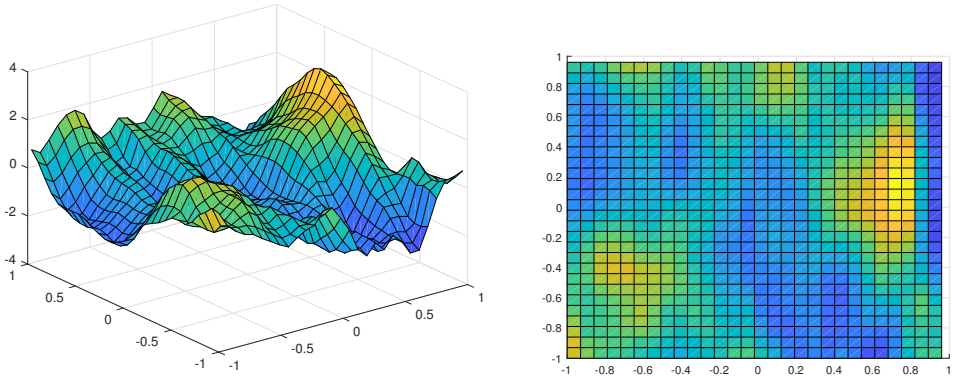


Figure 5.1: Non-stationary Matérn realisation with ARD covariance function.

This isotropic construction means that we assume the range of dependence of the non-stationary process to be the same in all directions. In such case, the non-stationary family of covariance functions in Eq. (5.3) can be written as,

$$C_{\boldsymbol{\phi}}^{\mathrm{NS}}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \tau_z^2 \frac{\ell(\boldsymbol{x}_i)^{\frac{D}{2}} \ell(\boldsymbol{x}_j)^{\frac{D}{2}}}{\left([\ell^2(\boldsymbol{x}_i) + \ell^2(\boldsymbol{x}_j)]/2\right)^{\frac{D}{2}}} R_\psi \left(\sqrt{\frac{\sum_{d=1}^{D} 2(x_{id} - x_{jd})^2}{\ell^2(\boldsymbol{x}_i) + \ell^2(\boldsymbol{x}_j)}} \right). \tag{5.5}$$

A realisation from a non-stationary Mátern covariance with scaled identity kernel matrices is shown in Figure 5.2. As illustrated in Figure 5.4(b), the assumed structure of the kernel matrices will produce circles (rather than ellipses).

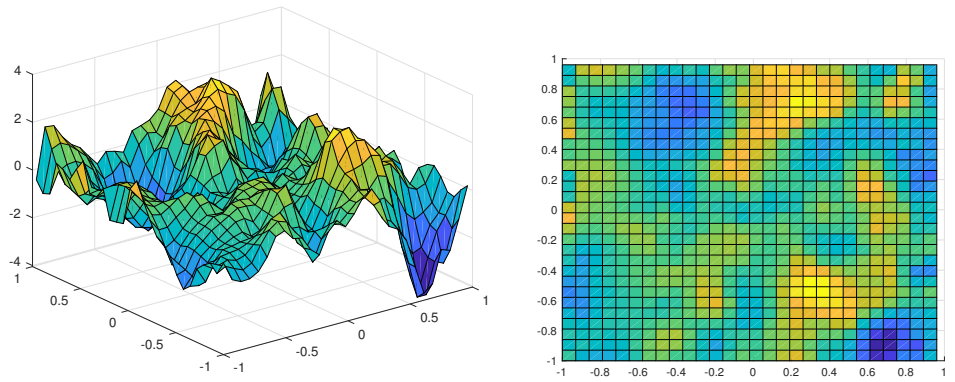


Figure 5.2: Non-stationary Matérn realisation with isotropic covariance matrix.

5.2.3 Separable ARD covariance

Lastly, we consider constructing a D-dimensional non-stationary covariance function as the product of D one-dimensional non-stationary covariance functions. This results in what is termed as a separable covariance function (see Chapter 2, Section 2.1.2.3), such that $C^{\text{NS}}_{\boldsymbol{\phi}}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \prod_{d=1}^{D} C^{\text{NS}}_{\boldsymbol{\phi}_d}(x_{id}, x_{jd})$. The separability assumption can offer computational benefits (when a process is observed on a grid, this assumption reduces the computational complexity of the required matrix computations (Rougier, 2017)).

This family of non-stationary kernels can be written as

$$C^{\text{NS}}_{\boldsymbol{\phi}}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \tau_z^2 \prod_{d=1}^{D} \frac{\ell_d(x_{id})^{\frac{1}{2}} \ell_d(x_{jd})^{\frac{1}{2}}}{\left([\ell_d^2(x_{id}) + \ell_d^2(x_{jd})]/2\right)^{\frac{1}{2}}} R_\psi \left(\sqrt{\frac{2(x_{id} - x_{jd})^2}{\ell^2(x_{id}) + \ell^2(x_{jd})}} \right). \quad (5.6)$$

Here, we are assuming a single magnitude parameter τ_z^2 and the same correlation function parameter ψ; however, it is also possible to define dimension-specific parameters. At first instance, this setting may appear equivalent to a non-stationary ARD kernel (Section 5.2.1). However, note that in contrast to the non-stationary ARD kernel, in this setting the stationary correlation function is required to factorise across dimensions, i.e.

$$R_\psi \left(\sqrt{\sum_{d=1}^{D} \frac{2(x_{id} - x_{jd})^2}{\ell_d^2(x_{id}) + \ell_d^2(x_{jd})}} \right) = \prod_{d=1}^{D} R_\psi \left(\sqrt{\frac{2(x_{id} - x_{jd})^2}{\ell^2(x_{id}) + \ell^2(x_{jd})}} \right).$$

For some choices of the correlation function, the two formulations are indeed equivalent, e.g. for the squared exponential and the exponential correlation functions.

Figure 5.3 depicts a two-dimensional realisation from a non-stationary Matérn process with a separable ARD kernel. The kernel matrices in this case correpond to those illustrated in Figure 5.4(a).

5.3 Non-stationary variationally sparse MCMC

We present the derivations required to obtain the optimal sparse variational distributions for each of the formulations of the kernel matrices discussed in Section 5.2. In addition, we provide detailed pseudo-code to implement the methodology. While this approach can be extended for other GP models, e.g. to accommodate other response types, the focus of this thesis is on non-stationary GP regression models,

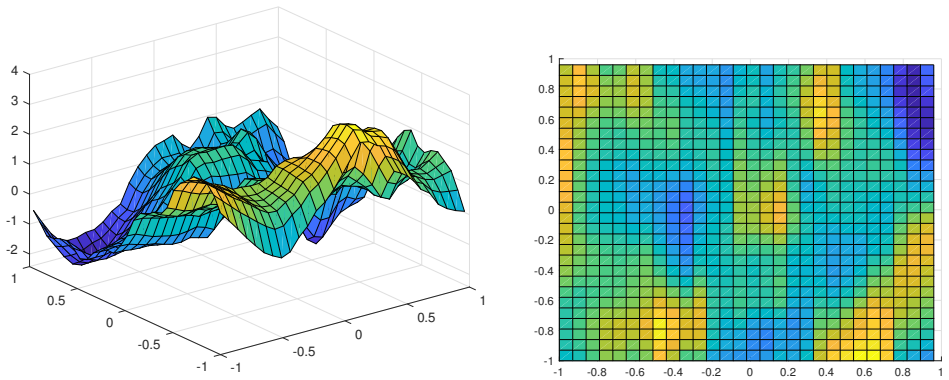
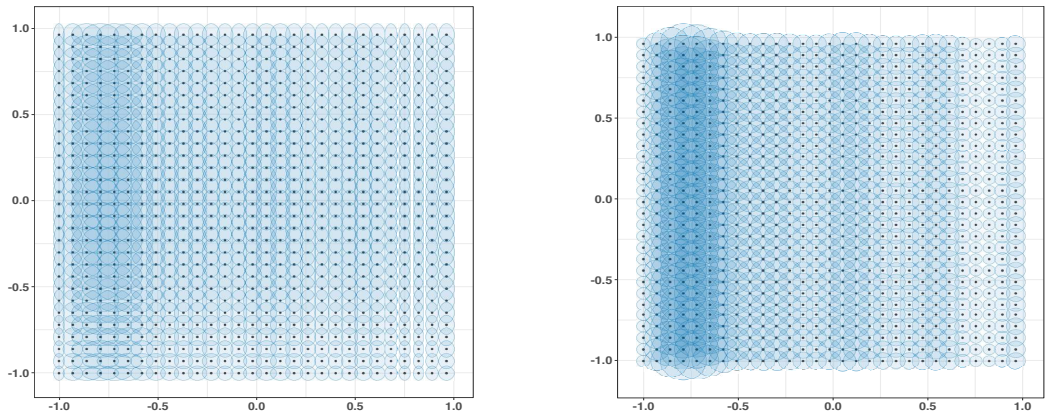


Figure 5.3: Non-stationary Mátern realisations with separable ARD covariance function.



(a) Diagonal (b) Scaled identity

Figure 5.4: Kernel matrices. (a): Depicts axis aligned ellipses formed by $\Sigma(\boldsymbol{x}_n) = \text{diag}(\ell_1(x_{n1}), \ell_2(x_{n2}))$ (b): Illustrates circles of different diameters obtained through $\Sigma(\boldsymbol{x}_n) = \ell^2(\boldsymbol{x}_n)I_2$. Ellipses and circles are scaled for ease of visualisation.

and therefore, we concentrate on this case. We assume a zero-centred non-stationary Gaussian process prior on the unknown function $z(\cdot)$ with the non-stationary covariance functions in Eq. (5.4), (5.5), or (5.6). The model assumes independence across all the parameters, those in the non-stationary covariance, $\boldsymbol{\phi}$, and the noise variance, σ_ε^2.

5.3.1 Optimal sparse variational distributions

5.3.1.1 ARD case

The kernel matrices have the form $\Sigma(\cdot) := \text{diag}(\ell_1^2(\cdot), \dots, \ell_D^2(\cdot))$, which results in the covariance function from Eq. (5.4). As $\ell_d(\cdot) > 0$, we consider a log transformation $u_d(\cdot) \coloneqq \log(\ell_d(\cdot))$. The independence assumption across parameters, results in the joint prior:

$$\pi(\sigma_\varepsilon^2, \tau_z^2, \psi, u_1(\cdot), \dots, u_D(\cdot)) = \pi(\sigma_\varepsilon^2)\pi(\tau_z^2)\pi(\psi) \prod_{d=1}^{D} \pi(u_d(\cdot)).$$

We assign a constant-mean stationary GP prior for each $u_d(\cdot) \sim \text{GP}(\mu_{u_d}, C^{\text{S}}_{\boldsymbol{\varphi}_d}(\cdot,\cdot))$. Note that each of the required stationary covariance functions has its own set of hyperparameters, $\boldsymbol{\varphi}_d$. Consequently, we specify a prior for each of the parameters $\boldsymbol{\varphi}_d$, with $d = 1, \dots, D$.

Under this setting, the posterior of interest is

$$\begin{aligned}\pi(\boldsymbol{z}, \boldsymbol{u}_{:1}, \dots, \boldsymbol{u}_{:d}, \sigma_\varepsilon^2, \tau_z^2, \psi, \boldsymbol{\varphi}_1, \dots, \boldsymbol{\varphi}_D \mid \boldsymbol{y}, X) \propto \text{N}(\boldsymbol{y} \mid \boldsymbol{z}, \sigma_\varepsilon^2 I_N)\pi(\sigma_\varepsilon^2)\pi(\tau_z^2)\pi(\psi) \\ \pi(\boldsymbol{z} \mid \boldsymbol{u}_{:1}, \dots, \boldsymbol{u}_{:d}, \tau_z^2, \psi) \prod_{d=1}^{D} \pi(\boldsymbol{u}_{:d} \mid \boldsymbol{\varphi}_d)\pi(\boldsymbol{\varphi}_d),\end{aligned}$$

where $\boldsymbol{u}_{:d} = (u_d(x_{1d}), \dots, u_d(x_{Nd}))^T$, for $d = 1, \dots, D$, and $\boldsymbol{z} = (z_1, \dots z_N)^T$. For notational convenience, let us define U to be a matrix with columns $\boldsymbol{u}_{:d} = (u_d(x_{1d}), \dots, u_d(x_{Nd}))^T$ for $d = 1, \dots D$ and rows $\boldsymbol{u}_{n:} = (u_1(x_{n1}), \dots u_D(x_{nD}))$ for $n = 1, \dots N$. We also define, $\boldsymbol{\varphi} \coloneqq (\boldsymbol{\varphi}_1, \dots, \boldsymbol{\varphi}_D)$, and $\boldsymbol{\theta} \coloneqq (\sigma_\varepsilon^2, \tau_z^2, \psi)$.

Following the sparse framework of Snelson and Ghahramani (2006), we augment the model with a set of inducing points $\widetilde{X} = (\widetilde{\boldsymbol{x}}_1, \dots, \widetilde{\boldsymbol{x}}_M)^T$ with $M \ll N$ and collect the values of the latent functions at the inducing points into the vector $\widetilde{\boldsymbol{z}} = (z(\widetilde{\boldsymbol{x}}_1), \dots, z(\widetilde{\boldsymbol{x}}_M))^T$. Additionally, we define $\widetilde{U}$ as a matrix of size $(M \times D)$ collecting the values of the log length-scale process at the inducing points, with

elements $u_d(\widetilde{x}_{md})$, columns $\widetilde{\boldsymbol{u}}_{:d}$, and rows $\widetilde{\boldsymbol{u}}_{m:}$. From properties of the GP, we have

$$\begin{aligned}\pi(U, \widetilde{U} \mid \boldsymbol{\varphi}) &= \prod_{d=1}^{D} \pi(\boldsymbol{u}_{:d} \mid \widetilde{\boldsymbol{u}}_{:d}, \boldsymbol{\varphi}_d)\pi(\widetilde{\boldsymbol{u}}_{:d} \mid \boldsymbol{\varphi}_d) \\ &= \prod_{d=1}^{D} \mathrm{N}(\boldsymbol{u}_{:d} \mid \boldsymbol{\mu}^*_{u_d}, \Omega_{u_d})\mathrm{N}(\widetilde{\boldsymbol{u}}_{:d} \mid \boldsymbol{\mu}_{u_d}, C^{\mathrm{S}}_{\widetilde{\boldsymbol{u}}_{:d},\widetilde{\boldsymbol{u}}_{:d}}),\end{aligned} \tag{5.7}$$

with $\boldsymbol{\mu}_{u_d}$ an M-dimensional vector with elements μ_{u_d}, the mean $\boldsymbol{\mu}^*_{u_d} = \boldsymbol{\mu}_{u_d} + C^{\mathrm{S}}_{\boldsymbol{u}_{:d},\widetilde{\boldsymbol{u}}_{:d}}(C^{\mathrm{S}}_{\widetilde{\boldsymbol{u}}_{:d},\widetilde{\boldsymbol{u}}_{:d}})^{-1}(\widetilde{\boldsymbol{u}}_{:d} - \boldsymbol{\mu}_{u_d})$, and variance $\Omega_{u_d} = C^{\mathrm{S}}_{\boldsymbol{u}_{:d},\boldsymbol{u}_{:d}} - C^{\mathrm{S}}_{\boldsymbol{u}_{:d},\widetilde{\boldsymbol{u}}_{:d}}(C^{\mathrm{S}}_{\widetilde{\boldsymbol{u}}_{:d},\widetilde{\boldsymbol{u}}_d})^{-1}C^{\mathrm{S}}_{\widetilde{\boldsymbol{u}}_{:d},\boldsymbol{u}_{:d}}$. Additionally,

$$\begin{aligned}\pi(\boldsymbol{z}, \widetilde{\boldsymbol{z}} \mid U, \widetilde{U}, \tau_z^2, \psi) &= \pi(\boldsymbol{z} \mid \widetilde{\boldsymbol{z}}, U, \widetilde{U}, \tau_z^2, \psi)\pi(\widetilde{\boldsymbol{z}} \mid \widetilde{U}, \tau_z^2, \psi) \\ &= \mathrm{N}(\boldsymbol{z} \mid \boldsymbol{\mu}^*_z, \Omega_z)\mathrm{N}(\widetilde{\boldsymbol{z}} \mid 0, C^{\mathrm{NS}}_{\widetilde{\boldsymbol{z}},\widetilde{\boldsymbol{z}}}),\end{aligned} \tag{5.8}$$

where $\boldsymbol{\mu}^*_z = C^{\mathrm{NS}}_{\boldsymbol{z},\widetilde{\boldsymbol{z}}}(C^{\mathrm{NS}}_{\widetilde{\boldsymbol{z}},\widetilde{\boldsymbol{z}}})^{-1}\widetilde{\boldsymbol{z}}$ and $\Omega_z = C^{\mathrm{NS}}_{\boldsymbol{z},\boldsymbol{z}} - C^{\mathrm{NS}}_{\boldsymbol{z},\widetilde{\boldsymbol{z}}}(C^{\mathrm{NS}}_{\widetilde{\boldsymbol{z}},\widetilde{\boldsymbol{z}}})^{-1}C^{\mathrm{NS}}_{\widetilde{\boldsymbol{z}},\boldsymbol{z}}$. In Eq. (5.7) we make use of the notation $C^{\mathrm{S}}_{\widetilde{\boldsymbol{u}}_{:d},\widetilde{\boldsymbol{u}}_{:d}}$ to denote the covariance matrix constructed by evaluating a stationary kernel at the d-dimension of the inducing inputs, and $C^{\mathrm{S}}_{\boldsymbol{u}_{:d},\widetilde{\boldsymbol{u}}_{:d}}$ to denote the stationary cross-covariance matrix between the d-dimension of the observed locations and the d-dimension of the inducing inputs. The matrices in Eq. (5.8) are similarly defined.

After augmentation, the posterior of the latent variables, parameters and hyper-parameters is given by:

$$\begin{aligned}\pi(\boldsymbol{z}, \widetilde{\boldsymbol{z}}, U, \widetilde{U}, \boldsymbol{\theta}, \boldsymbol{\varphi} \mid \boldsymbol{y}, X, \widetilde{X}) \propto \mathrm{N}(\boldsymbol{y} \mid \boldsymbol{z}, \sigma_\varepsilon^2 I_N)\pi(\boldsymbol{z} \mid \widetilde{\boldsymbol{z}}, U, \widetilde{U}, \tau_z^2, \psi)\pi(\widetilde{\boldsymbol{z}} \mid \widetilde{U}, \tau_z^2, \psi) \\ \pi(\boldsymbol{\theta}) \left(\prod_{d=1}^{D} \pi(\boldsymbol{u}_{:d} \mid \widetilde{\boldsymbol{u}}_{:d}, \boldsymbol{\varphi}_d)\pi(\widetilde{\boldsymbol{u}}_{:d} \mid \boldsymbol{\varphi}_d)\pi(\boldsymbol{\varphi}_d) \right).\end{aligned}$$

We assume that the approximate variational posterior takes the form:

$$q(\boldsymbol{z}, \widetilde{\boldsymbol{z}}, U, \widetilde{U}, \boldsymbol{\theta}, \boldsymbol{\varphi}) = \pi(\boldsymbol{z} \mid \widetilde{\boldsymbol{z}}, U, \widetilde{U}, \tau_z^2, \psi) \left(\prod_{d=1}^{D} \pi(\boldsymbol{u}_{:d} \mid \widetilde{\boldsymbol{u}}_{:d}, \boldsymbol{\varphi}_d) \right) q(\widetilde{\boldsymbol{z}}, \widetilde{U}, \boldsymbol{\theta}, \boldsymbol{\varphi}). \tag{5.9}$$

We then seek the variational posterior distribution which minimizes the KL diver-

gence to the true posterior; that is

$$\begin{aligned}
&\mathrm{KL}(q(\boldsymbol{z}, \widetilde{\boldsymbol{z}}, U, \widetilde{U}, \boldsymbol{\theta}, \boldsymbol{\varphi}) \parallel \pi(\boldsymbol{z}, \widetilde{\boldsymbol{z}}, U, \widetilde{U}, \boldsymbol{\theta}, \boldsymbol{\varphi} \mid \boldsymbol{y}, X, \widetilde{X})) \\
&= -\mathbb{E}_q \left[\log \left(\frac{\pi(\boldsymbol{z}, \widetilde{\boldsymbol{z}}, U, \widetilde{U}, \boldsymbol{\theta}, \boldsymbol{\varphi} \mid \boldsymbol{y}, X, \widetilde{X})}{q(\boldsymbol{z}, \widetilde{\boldsymbol{z}}, U, \widetilde{U}, \boldsymbol{\theta}, \boldsymbol{\varphi})} \right) \right] \\
&= -\mathbb{E}_q \left[\log \left(\frac{\mathrm{N}(\boldsymbol{y} \mid \boldsymbol{z}, \sigma_\varepsilon^2 I_N) \pi(\widetilde{\boldsymbol{z}} \mid \widetilde{U}, \tau_z^2, \psi) \left(\prod_{d=1}^D \pi(\widetilde{\boldsymbol{u}}_{:d} \mid \boldsymbol{\varphi}_d) \pi(\boldsymbol{\varphi}_d) \right) \pi(\boldsymbol{\theta})}{q(\widetilde{\boldsymbol{z}}, \widetilde{U}, \boldsymbol{\theta}, \boldsymbol{\varphi})} \right) \right] \\
&\quad + \log(p(\boldsymbol{y} \mid X)),
\end{aligned} \tag{5.10}$$

where the expectation is taken with respect to $q(\boldsymbol{z}, \widetilde{\boldsymbol{z}}, U, \widetilde{U}, \boldsymbol{\theta}, \boldsymbol{\varphi})$. By using the assumed decomposition in Eq. (5.9) and first taking the inner expectation with respect to $\boldsymbol{z}$ and U, Eq. (5.10) can be written as:

$$\begin{aligned}
-\mathbb{E}_{q(\widetilde{\boldsymbol{z}}, \widetilde{U}, \boldsymbol{\theta}, \boldsymbol{\varphi})} \Bigg[\log \Bigg(& \frac{\exp\left(\mathbb{E}_{\pi(\boldsymbol{z}, U \mid \widetilde{\boldsymbol{z}}, \widetilde{U}, \boldsymbol{\theta}, \boldsymbol{\varphi})} [\log(\mathrm{N}(\boldsymbol{y} \mid \boldsymbol{z}, \sigma_\varepsilon^2 I_N))] \right) \pi(\widetilde{\boldsymbol{z}} \mid \widetilde{U}, \tau_z^2, \psi)}{q(\widetilde{\boldsymbol{z}}, \widetilde{U}, \boldsymbol{\theta}, \boldsymbol{\varphi})} \Bigg) \\
& + \log \left(\frac{\left(\prod_{d=1}^D \pi(\widetilde{\boldsymbol{u}}_{:d} \mid \boldsymbol{\varphi}_d) \pi(\boldsymbol{\varphi}_d) \right) \pi(\boldsymbol{\theta})}{q(\widetilde{\boldsymbol{z}}, \widetilde{U}, \boldsymbol{\theta}, \boldsymbol{\varphi})} \right) \Bigg] + \log(p(\boldsymbol{y} \mid X)).
\end{aligned}$$

Minimisation reveals that the optimal variational posterior corresponds to

$$\begin{aligned}
q(\widetilde{\boldsymbol{z}}, \widetilde{U}, \boldsymbol{\theta}, \boldsymbol{\varphi}) \propto \exp & \left(\mathbb{E}_{\pi(\boldsymbol{z}, U \mid \widetilde{\boldsymbol{z}}, \widetilde{U}, \boldsymbol{\theta}, \boldsymbol{\varphi})} [\log(\mathrm{N}(\boldsymbol{y} \mid \boldsymbol{z}, \sigma_\varepsilon^2 I_N))] \right) \pi(\widetilde{\boldsymbol{z}} \mid \widetilde{U}, \tau_z^2, \psi) \\
& \left(\prod_{d=1}^D \pi(\widetilde{\boldsymbol{u}}_{:d} \mid \boldsymbol{\varphi}_d) \pi(\boldsymbol{\varphi}_d) \right) \pi(\boldsymbol{\theta}).
\end{aligned} \tag{5.11}$$

The expected log-likelihood term in Eq. (5.11) is

$$\begin{aligned}
\mathbb{E}_{\pi(\boldsymbol{z}, U \mid \widetilde{\boldsymbol{z}}, \widetilde{U}, \boldsymbol{\theta}, \boldsymbol{\varphi})} [\log(\mathrm{N}(\boldsymbol{y} \mid \boldsymbol{z}, \sigma_\varepsilon^2 I_N))] = & -\frac{N}{2} \log(2\pi\sigma_\varepsilon^2) - \frac{1}{2\sigma_\varepsilon^2} \left(\sum_{n=1}^N \mathbb{V}_{\pi(z_n, \boldsymbol{u}_{n:} \mid \widetilde{\boldsymbol{z}}, \widetilde{U}, \boldsymbol{\theta}, \boldsymbol{\varphi})} [z_n] \right) \\
& - \frac{1}{2\sigma_\varepsilon^2} \left(\sum_{n=1}^N (y_n - \mathbb{E}_{\pi(z_n, \boldsymbol{u}_{n:} \mid \widetilde{\boldsymbol{z}}, \widetilde{U} \boldsymbol{\theta}, \boldsymbol{\varphi})} [z_n])^2 \right),
\end{aligned}$$

where by taking the expectation with respect to $\pi(z_n \mid \widetilde{\boldsymbol{z}}, U, \widetilde{U}, \tau_z^2, \psi)$ (see Eq. (5.8)) we have

$$\mathbb{E}_{\pi(z_n, \boldsymbol{u}_{n:} \mid \widetilde{\boldsymbol{z}}, \widetilde{U}, \boldsymbol{\theta}, \boldsymbol{\varphi})} [z_n] = \mathbb{E}_{\pi(\boldsymbol{u}_{n:} \mid \widetilde{U}, \boldsymbol{\theta}, \boldsymbol{\varphi})} [C^{\mathrm{NS}}_{z_n, \widetilde{\boldsymbol{z}}}] (C^{\mathrm{NS}}_{\widetilde{\boldsymbol{z}}, \widetilde{\boldsymbol{z}}})^{-1} \widetilde{\boldsymbol{z}},$$

$$\mathbb{V}_{\pi(z_n,\boldsymbol{u}_{n:}|\widetilde{\boldsymbol{z}},\widetilde{U},\boldsymbol{\theta},\boldsymbol{\varphi})}[z_n] = \mathbb{E}_{\pi(\boldsymbol{u}_{n:}|\widetilde{U},\boldsymbol{\theta},\boldsymbol{\varphi})}[\mathbb{V}_{\pi(z_n|U,\widetilde{\boldsymbol{z}},\widetilde{U},\boldsymbol{\theta},\boldsymbol{\varphi})}(z_n)] + \mathbb{V}_{\pi(\boldsymbol{u}_{n:}|\widetilde{U},\boldsymbol{\theta},\boldsymbol{\varphi})}[\mathbb{E}_{\pi(z_n|U,\widetilde{\boldsymbol{z}},\widetilde{U},\boldsymbol{\theta},\boldsymbol{\varphi})}(z_n)].$$

In particular, the two terms in the variance are

$$\mathbb{E}_{\pi(\boldsymbol{u}_{n:}|\widetilde{U},\boldsymbol{\theta},\boldsymbol{\varphi})}[\mathbb{V}_{\pi(z_n|U,\widetilde{\boldsymbol{z}},\widetilde{U},\boldsymbol{\theta},\boldsymbol{\varphi})}(z_n)] = \tau_z^2 - \mathbb{E}_{\pi(\boldsymbol{u}_{n:}|\widetilde{U},\boldsymbol{\theta},\boldsymbol{\varphi})}[C^{\mathrm{NS}}_{z_n,\widetilde{\boldsymbol{z}}}(C^{\mathrm{NS}}_{\widetilde{\boldsymbol{z}},\widetilde{\boldsymbol{z}}})^{-1}C^{\mathrm{NS}}_{\widetilde{\boldsymbol{z}},z_n}],$$

$$\mathbb{V}_{\pi(\boldsymbol{u}_{n:}|\widetilde{U},\boldsymbol{\theta},\boldsymbol{\varphi})}[\mathbb{E}_{\pi(z_n|U,\widetilde{\boldsymbol{z}},\widetilde{U},\boldsymbol{\theta},\boldsymbol{\varphi})}(z_n)] = \widetilde{\boldsymbol{z}}^T(C^{\mathrm{NS}}_{\widetilde{\boldsymbol{z}},\widetilde{\boldsymbol{z}}})^{-1}\mathbb{V}_{\pi(\boldsymbol{u}_{n:}|\widetilde{U},\boldsymbol{\theta},\boldsymbol{\varphi})}[C^{\mathrm{NS}}_{z_n,\widetilde{\boldsymbol{z}}}](C^{\mathrm{NS}}_{\widetilde{\boldsymbol{z}},\widetilde{\boldsymbol{z}}})^{-1}\widetilde{\boldsymbol{z}}.$$

While $\boldsymbol{z}$ can marginalised analytically, the expectations with respect to $\pi(\boldsymbol{u}_{n:}|\widetilde{U},\boldsymbol{\theta},\boldsymbol{\varphi})$ are intractable; we denote the required expectations by

$$\begin{aligned}
\boldsymbol{\beta}_n &= \mathbb{E}_{\pi(\boldsymbol{u}_{n:}|\widetilde{U},\boldsymbol{\theta},\boldsymbol{\varphi})}[C^{\mathrm{NS}}_{z_n,\widetilde{\boldsymbol{z}}}],\\
\alpha_n &= \mathbb{E}_{\pi(\boldsymbol{u}_{n:}|\widetilde{U},\boldsymbol{\theta},\boldsymbol{\varphi})}[C^{\mathrm{NS}}_{z_n,\widetilde{\boldsymbol{z}}}(C^{\mathrm{NS}}_{\widetilde{\boldsymbol{z}},\widetilde{\boldsymbol{z}}})^{-1}C^{\mathrm{NS}}_{\widetilde{\boldsymbol{z}},z_n}],\\
P_n &= \mathbb{E}_{\pi(\boldsymbol{u}_{n:}|\widetilde{U},\boldsymbol{\theta},\boldsymbol{\varphi})}\left[C^{\mathrm{NS}}_{\widetilde{\boldsymbol{z}},z_n}C^{\mathrm{NS}}_{z_n,\widetilde{\boldsymbol{z}}}\right],
\end{aligned} \tag{5.12}$$

Then, we have that the variational posterior is given by

$$q(\widetilde{\boldsymbol{z}},\widetilde{U},\boldsymbol{\theta},\boldsymbol{\varphi}) \propto \left(\frac{\sigma_\varepsilon^{-2}}{2\pi}\right)^{\frac{N}{2}} \pi(\widetilde{\boldsymbol{z}}|\widetilde{U},\tau_z^2,\psi)\pi(\boldsymbol{\theta})\left(\prod_{d=1}^{D}\pi(\widetilde{\boldsymbol{u}}_{:d}\mid\boldsymbol{\varphi}_d)\pi(\boldsymbol{\varphi}_d)\right)\times$$
$$\exp\left(-\frac{1}{2\sigma_\varepsilon^2}\sum_{n=1}^{N}(y_n-\boldsymbol{\beta}_n(C^{\mathrm{NS}}_{\widetilde{\boldsymbol{z}},\widetilde{\boldsymbol{z}}})^{-1}\widetilde{\boldsymbol{z}})^2+\tau_z^2-\alpha_n+\widetilde{\boldsymbol{z}}^T(C^{\mathrm{NS}}_{\widetilde{\boldsymbol{z}},\widetilde{\boldsymbol{z}}})^{-1}(P_n-\boldsymbol{\beta}_n^T\boldsymbol{\beta}_n)(C^{\mathrm{NS}}_{\widetilde{\boldsymbol{z}},\widetilde{\boldsymbol{z}}})^{-1}\widetilde{\boldsymbol{z}}\right). \tag{5.13}$$

We intend to explore the variational posterior in Eq. (5.13) using MCMC, but high correlations between $\widetilde{\boldsymbol{z}}$ and $(\widetilde{U},\boldsymbol{\theta},\boldsymbol{\varphi})$ will result in poor mixing in a Gibbs sampling framework. However, the latent variables $\widetilde{\boldsymbol{z}}$ can be marginalised. Specifically, the marginal variational posterior of $(\widetilde{U},\boldsymbol{\theta},\boldsymbol{\varphi})$ is

$$q(\widetilde{U},\boldsymbol{\theta},\boldsymbol{\varphi}) \propto \sigma_\varepsilon^{-N}\pi(\boldsymbol{\theta})\left(\prod_{d=1}^{D}\pi(\widetilde{\boldsymbol{u}}_{:d}\mid\boldsymbol{\varphi}_d)\pi(\boldsymbol{\varphi}_d)\right)\exp\left(\frac{1}{2\sigma_\varepsilon^4}\boldsymbol{y}^T B\left(C^{\mathrm{NS}}_{\widetilde{\boldsymbol{z}},\widetilde{\boldsymbol{z}}}+\sigma_\varepsilon^{-2}\boldsymbol{P}\right)^{-1}B^T\boldsymbol{y}\right)$$
$$|C^{\mathrm{NS}}_{\widetilde{\boldsymbol{z}},\widetilde{\boldsymbol{z}}}+\sigma_\varepsilon^{-2}\boldsymbol{P}|^{-\frac{1}{2}}|C^{\mathrm{NS}}_{\widetilde{\boldsymbol{z}},\widetilde{\boldsymbol{z}}}|^{\frac{1}{2}}\exp\left(-\frac{1}{2\sigma_\varepsilon^2}\sum_{n=1}^{N}(y_n^2+\tau_z^2)+\frac{1}{2\sigma_\varepsilon^2}\sum_{i,j=1}^{M}\left((C^{\mathrm{NS}}_{\widetilde{\boldsymbol{z}},\widetilde{\boldsymbol{z}}})^{-1}\odot\boldsymbol{P}\right)_{ij}\right), \tag{5.14}$$

with B an $N\times M$ matrix with rows $\boldsymbol{\beta}_n$, $\boldsymbol{P}=\sum_{n=1}^{N}P_n$, and where $\odot$ denotes the Hadamard product. Note that $\sum_{n=1}^{N}\alpha_n$ in Eq. (5.13) is here rewritten as $\sum_{i,j=1}^{M}\left((C^{\mathrm{NS}}_{\widetilde{\boldsymbol{z}},\widetilde{\boldsymbol{z}}})^{-1}\odot\boldsymbol{P}\right)_{ij}$ to avoid the computation of N expectations. A detailed

derivation of Eq. 5.14 is provided in Appendix D.

An MCMC scheme can be devised to simulate from the marginal variational posterior in Eq. (5.14), and for a given sample $(\widetilde{U}, \boldsymbol{\theta}, \boldsymbol{\varphi})$, we can simulate $\widetilde{\boldsymbol{z}}$ from its conditional variational posterior, which is a multivariate Gaussian (see Appendix D):

$$\widetilde{\boldsymbol{z}} \mid \boldsymbol{\theta}, \boldsymbol{\varphi} \sim \mathrm{N}\left(\sigma_\varepsilon^{-2} C^{\mathrm{NS}}_{\widetilde{\boldsymbol{z}},\widetilde{\boldsymbol{z}}} \left(C^{\mathrm{NS}}_{\widetilde{\boldsymbol{z}},\widetilde{\boldsymbol{z}}} + \sigma_\varepsilon^{-2}\boldsymbol{P}\right)^{-1} B^T \boldsymbol{y}, C^{\mathrm{NS}}_{\widetilde{\boldsymbol{z}},\widetilde{\boldsymbol{z}}} \left(C^{\mathrm{NS}}_{\widetilde{\boldsymbol{z}},\widetilde{\boldsymbol{z}}} + \sigma_\varepsilon^{-2}\boldsymbol{P}\right)^{-1} C^{\mathrm{NS}}_{\widetilde{\boldsymbol{z}},\widetilde{\boldsymbol{z}}}\right).$$

High correlations still exist between the latent variables $\widetilde{\boldsymbol{u}}_{:d}$ and its corresponding parameters, $\boldsymbol{\varphi}_d$. To overcome this, we employ *whitening*. We define $\widetilde{\boldsymbol{u}}_{:d} = L(\boldsymbol{\varphi}_d)\widetilde{\boldsymbol{\zeta}}_d + \boldsymbol{\mu}_{u_d}$ with $\widetilde{\boldsymbol{\zeta}}_d \sim \mathrm{N}(0, I_M)$ and $L(\boldsymbol{\varphi}_d)L(\boldsymbol{\varphi}_d)^T = C^{\mathrm{S}}_{\widetilde{\boldsymbol{u}}_{:d},\widetilde{\boldsymbol{u}}_{:d}}$, for $d = 1, \ldots, D$. MCMC methods are used to simulate from the whitened marginal variational posterior:

$$q(\widetilde{\boldsymbol{\zeta}}_1, \ldots, \widetilde{\boldsymbol{\zeta}}_D, \boldsymbol{\theta}, \boldsymbol{\varphi}) \propto \left(\prod_{d=1}^{D} \mathrm{N}(\widetilde{\boldsymbol{\zeta}}_d \mid 0, I_M)\pi(\boldsymbol{\varphi}_d)\right) \pi(\boldsymbol{\theta})\sigma_\varepsilon^{-N} \exp\left(-\frac{1}{2\sigma_\varepsilon^2}\sum_{n=1}^{N}(y_n^2) - \frac{N\tau_z^2}{2\sigma_\varepsilon^2}\right)$$
$$\left[\exp\left(\frac{1}{2\sigma_\varepsilon^2}\sum_{i,j=1}^{M}\left((C^{\mathrm{NS}}_{\widetilde{\boldsymbol{z}},\widetilde{\boldsymbol{z}}})^{-1} \odot \boldsymbol{P}\right)_{ij} + \frac{1}{2\sigma_\varepsilon^4}\boldsymbol{y}^T B\left(C^{\mathrm{NS}}_{\widetilde{\boldsymbol{z}},\widetilde{\boldsymbol{z}}} + \sigma_\varepsilon^{-2}\boldsymbol{P}\right)^{-1} B^T \boldsymbol{y}\right) \frac{|C^{\mathrm{NS}}_{\widetilde{\boldsymbol{z}},\widetilde{\boldsymbol{z}}}|^{\frac{1}{2}}}{|C^{\mathrm{NS}}_{\widetilde{\boldsymbol{z}},\widetilde{\boldsymbol{z}}} + \sigma_\varepsilon^{-2}\boldsymbol{P}|^{\frac{1}{2}}}\right]_{\widetilde{U}}, \tag{5.15}$$

where each $\widetilde{\boldsymbol{u}}_{:d}$ needed for the non-stationary kernel is replaced by $L(\boldsymbol{\varphi}_d)\widetilde{\boldsymbol{\zeta}}_d + \boldsymbol{\mu}_{u_d}$ in the expression in square brackets.

Note that the posterior of interest in Eq. (5.15), involves the computation of multivariate expectations $\boldsymbol{\beta}_n$ and P_n, for $n = 1, \ldots N$, shown in Eq. (5.12). Therefore, a suitable method to approximate them is required. In the special case when a non-stationary squared exponential or exponential kernel is employed in Eq. (5.4), $\boldsymbol{\beta}_n$ and P_n will be the product of D univariate expectations with respect to Gaussian random variables, which can each be approximated using Gauss-Hermite quadrature.

5.3.1.2 Isotropic case

The kernel matrices have the form $\Sigma(\cdot) := \ell^2(\cdot)I_D$, resulting in the covariance function in Eq. (5.5). In this case, a single GP is required to parametrise the kernel matrices. Specifically, we use a constant mean stationary GP prior over the log transformed length-scale process; that is, $u(\cdot) := \log(\ell(\cdot)) \sim \mathrm{GP}(\mu_u, C^{\mathrm{S}}_{\boldsymbol{\varphi}}(\cdot,\cdot))$. The joint prior over the parameters and hyperparameters is $\pi(\sigma_\varepsilon^2, \tau_z^2, \psi, u(\cdot), \boldsymbol{\varphi}) = \pi(\sigma_\varepsilon^2)\pi(\tau_z^2)\pi(\psi)\pi(u(\cdot)|\boldsymbol{\varphi})\pi(\boldsymbol{\varphi})$. Therefore, the posterior of interest is

$$\pi(\boldsymbol{z}, \boldsymbol{u}, \sigma_\varepsilon^2, \tau_z^2, \psi, \boldsymbol{\varphi} \mid \boldsymbol{y}, X) \propto \mathrm{N}(\boldsymbol{y} \mid \boldsymbol{z}, \sigma_\varepsilon^2 I_N)\pi(\boldsymbol{z} \mid \boldsymbol{u}, \tau_z^2, \psi)\pi(\sigma_\varepsilon^2)\pi(\tau_z^2)\pi(\psi)\pi(\boldsymbol{u} \mid \boldsymbol{\varphi})\pi(\boldsymbol{\varphi}),$$

where $\boldsymbol{u} = (u_1, \ldots, u_N)^T$ with $u_n \equiv u(\boldsymbol{x}_n)$, and $\boldsymbol{z} = (z_1, \ldots, z_N)^T$.

Again, we augment the model with inducing points $\widetilde{\boldsymbol{x}}_m$, $m = 1, \ldots, M$, and define the vector $\widetilde{\boldsymbol{z}} = (z(\widetilde{\boldsymbol{x}}_1), \ldots, z(\widetilde{\boldsymbol{x}}_M)^T$ to be the latent function at the inducing locations. However, in the isotropic setting, as opposed to Section 5.3.1.1, we have only one latent vector $\widetilde{\boldsymbol{u}} = (u(\widetilde{\boldsymbol{x}}_1), \ldots, u(\widetilde{\boldsymbol{x}}_M))^T$ of the log length-scale process at the inducing locations. In this case, the augmented priors are

$$\pi(\boldsymbol{u}, \widetilde{\boldsymbol{u}} \mid \boldsymbol{\varphi}) = \pi(\boldsymbol{u} \mid \widetilde{\boldsymbol{u}}, \boldsymbol{\varphi})\pi(\widetilde{\boldsymbol{u}} \mid \boldsymbol{\varphi}) = \mathrm{N}(\boldsymbol{u} \mid \boldsymbol{\mu}_u^*, \Omega_u)\mathrm{N}(\widetilde{\boldsymbol{u}} \mid \boldsymbol{\mu}_u, C^{\mathrm{S}}_{\widetilde{\boldsymbol{u}},\widetilde{\boldsymbol{u}}}),$$

where $\boldsymbol{\mu}_u$ denotes an M-dimensional vector with entries μ_u, $\boldsymbol{\mu}_u^* = \boldsymbol{\mu}_u + C^{\mathrm{S}}_{\boldsymbol{u},\widetilde{\boldsymbol{u}}}(C^{\mathrm{S}}_{\widetilde{\boldsymbol{u}},\widetilde{\boldsymbol{u}}})^{-1}(\widetilde{\boldsymbol{u}} - \boldsymbol{\mu}_u)$, and $\Omega_u = C^{\mathrm{S}}_{\boldsymbol{u},\boldsymbol{u}} - C^{\mathrm{S}}_{\boldsymbol{u},\widetilde{\boldsymbol{u}}}(C^{\mathrm{S}}_{\widetilde{\boldsymbol{u}},\widetilde{\boldsymbol{u}}})^{-1}C^{\mathrm{S}}_{\widetilde{\boldsymbol{u}},\boldsymbol{u}}$, and

$$\pi(\boldsymbol{z}, \widetilde{\boldsymbol{z}} \mid \boldsymbol{u}, \widetilde{\boldsymbol{u}}, \tau_z^2, \psi) = \pi(\boldsymbol{z} \mid \widetilde{\boldsymbol{z}}, \boldsymbol{u}, \widetilde{\boldsymbol{u}}, \tau_z^2, \psi)\pi(\widetilde{\boldsymbol{z}} \mid \widetilde{\boldsymbol{u}}, \tau_z^2, \psi) = \mathrm{N}(\boldsymbol{z} \mid \boldsymbol{\mu}_z^*, \Omega_z)\mathrm{N}(\widetilde{\boldsymbol{z}} \mid 0, C^{\mathrm{NS}}_{\widetilde{\boldsymbol{z}},\widetilde{\boldsymbol{z}}}),$$

with $\boldsymbol{\mu}_z^* = C^{\mathrm{NS}}_{\boldsymbol{z},\widetilde{\boldsymbol{z}}}(C^{\mathrm{NS}}_{\widetilde{\boldsymbol{z}},\widetilde{\boldsymbol{z}}})^{-1}\widetilde{\boldsymbol{z}}$ and $\Omega_z = C^{\mathrm{NS}}_{\boldsymbol{z},\boldsymbol{z}} - C^{\mathrm{NS}}_{\boldsymbol{z},\widetilde{\boldsymbol{z}}}(C^{\mathrm{NS}}_{\widetilde{\boldsymbol{z}},\widetilde{\boldsymbol{z}}})^{-1}C^{\mathrm{NS}}_{\widetilde{\boldsymbol{z}},\boldsymbol{z}}$.

The augmented model results in the following posterior distribution

$$\begin{aligned}\pi(\boldsymbol{z}, \widetilde{\boldsymbol{z}}, \boldsymbol{u}, \widetilde{\boldsymbol{u}}, \boldsymbol{\theta}, \boldsymbol{\varphi} \mid \boldsymbol{y}, X, \widetilde{X}) \propto \mathrm{N}(\boldsymbol{y} \mid \boldsymbol{z}, \sigma_\varepsilon^2 I_N)\pi(\boldsymbol{z} \mid \widetilde{\boldsymbol{z}}, \boldsymbol{u}, \widetilde{\boldsymbol{u}}, \tau_z^2, \psi)\pi(\widetilde{\boldsymbol{z}} \mid \widetilde{\boldsymbol{u}}, \tau_z^2, \psi) \\ \pi(\boldsymbol{\theta})\pi(\boldsymbol{u} \mid \widetilde{\boldsymbol{u}}, \boldsymbol{\varphi})\pi(\widetilde{\boldsymbol{u}} \mid \boldsymbol{\varphi})\pi(\boldsymbol{\varphi}),\end{aligned}$$

where, as before, $\boldsymbol{\theta} \coloneqq \{\sigma_\varepsilon^2, \tau_z^2, \psi\}$. The approximate variational posterior is assumed to take the form:

$$q(\boldsymbol{z}, \widetilde{\boldsymbol{z}}, \boldsymbol{u}, \widetilde{\boldsymbol{u}}, \boldsymbol{\theta}, \boldsymbol{\varphi}) \propto \pi(\boldsymbol{z} \mid \widetilde{\boldsymbol{z}}, \boldsymbol{u}, \widetilde{\boldsymbol{u}}, \tau_z^2, \psi)\pi(\boldsymbol{u} \mid \widetilde{\boldsymbol{u}}, \boldsymbol{\varphi})q(\widetilde{\boldsymbol{z}}, \widetilde{\boldsymbol{u}}, \boldsymbol{\theta}, \boldsymbol{\varphi}).$$

Following the derivations from Section 5.3.1.1 (see Eq. (5.10)), we minimise the KL divergence: $\mathrm{KL}(q(\boldsymbol{z}, \widetilde{\boldsymbol{z}}, \boldsymbol{u}, \widetilde{\boldsymbol{u}}, \boldsymbol{\theta}, \boldsymbol{\varphi}) \parallel \pi(\boldsymbol{z}, \widetilde{\boldsymbol{z}}, \boldsymbol{u}, \widetilde{\boldsymbol{u}}, \boldsymbol{\theta}, \boldsymbol{\varphi} \mid \boldsymbol{y}, X, \widetilde{X}))$ to obtain the optimal variational posterior for the isotropic case

$$\begin{aligned}q(\widetilde{\boldsymbol{z}}, \widetilde{\boldsymbol{u}}, \boldsymbol{\theta}, \boldsymbol{\varphi}) \propto \exp\left(\mathbb{E}_{\pi(\boldsymbol{z},\boldsymbol{u}\mid\widetilde{\boldsymbol{z}},\widetilde{\boldsymbol{u}},\boldsymbol{\theta},\boldsymbol{\varphi})}[\log(\mathrm{N}(\boldsymbol{y} \mid \boldsymbol{z}, \sigma_\varepsilon^2 I_N))]\right)\pi(\widetilde{\boldsymbol{z}} \mid \widetilde{\boldsymbol{u}}, \tau_z^2, \psi) \\ \pi(\widetilde{\boldsymbol{u}} \mid \boldsymbol{\varphi})\pi(\boldsymbol{\varphi})\pi(\boldsymbol{\theta}).\end{aligned} \tag{5.16}$$

In contrast to the ARD covariance function, the expected log-likelihood only contains intractable univariate integrals, which are expectations with respect to univariate Gaussian random variables for any kernel employed. Therefore, these can be approximated using Gauss-Hermite quadrature (see Section 5.3.2 for an explanation

of this method). In this case:

$$\begin{aligned}\boldsymbol{\beta}_n &= \mathbb{E}_{\pi(u_n|\widetilde{\boldsymbol{u}},\boldsymbol{\theta},\boldsymbol{\varphi})}\left[C^{\mathrm{NS}}_{z_n,\widetilde{\boldsymbol{z}}}\right],\\ \alpha_n &= \mathbb{E}_{\pi(u_n|\widetilde{\boldsymbol{u}},\boldsymbol{\theta},\boldsymbol{\varphi})}\left[C^{\mathrm{NS}}_{z_n,\widetilde{\boldsymbol{z}}}(C^{\mathrm{NS}}_{\widetilde{\boldsymbol{z}},\widetilde{\boldsymbol{z}}})^{-1}C^{\mathrm{NS}}_{\widetilde{\boldsymbol{z}},z_n}\right],\\ P_n &= \mathbb{E}_{\pi(u_n|\widetilde{\boldsymbol{u}},\boldsymbol{\theta},\boldsymbol{\varphi})}\left[C^{\mathrm{NS}}_{\widetilde{\boldsymbol{z}},z_n}C^{\mathrm{NS}}_{z_n,\widetilde{\boldsymbol{z}}}\right],\end{aligned} \tag{5.17}$$

and, similarly, the variational posterior can be shown to have the form:

$$q(\widetilde{\boldsymbol{z}},\widetilde{\boldsymbol{u}},\boldsymbol{\theta},\boldsymbol{\varphi}) \propto \left(\frac{\sigma_\varepsilon^{-2}}{2\pi}\right)^{\frac{N}{2}} \pi(\widetilde{\boldsymbol{z}} \mid \widetilde{\boldsymbol{u}},\tau_z^2,\psi)\pi(\boldsymbol{\theta})\pi(\widetilde{\boldsymbol{u}} \mid \boldsymbol{\varphi})\pi(\boldsymbol{\varphi})\times$$
$$\exp\left(\frac{-1}{2\sigma_\varepsilon^2}\sum_{n=1}^{N}\left((y_n-\boldsymbol{\beta}_n(C^{\mathrm{NS}}_{\widetilde{\boldsymbol{z}},\widetilde{\boldsymbol{z}}})^{-1}\widetilde{\boldsymbol{z}})^2+\tau_z^2-\alpha_n+\widetilde{\boldsymbol{z}}^T(C^{\mathrm{NS}}_{\widetilde{\boldsymbol{z}},\widetilde{\boldsymbol{z}}})^{-1}(P_n-\boldsymbol{\beta}_n^T\boldsymbol{\beta}_n)(C^{\mathrm{NS}}_{\widetilde{\boldsymbol{z}},\widetilde{\boldsymbol{z}}})^{-1}\widetilde{\boldsymbol{z}}\right)\right).$$

As before, we suggest employing whitening to break the correlation between $\widetilde{\boldsymbol{u}}$ and $\boldsymbol{\varphi}$, defining $\widetilde{\boldsymbol{u}} = L(\boldsymbol{\varphi})\widetilde{\boldsymbol{\zeta}} + \boldsymbol{\mu}_u$, where $\widetilde{\boldsymbol{\zeta}} \sim \mathrm{N}(0, I_M)$ and $L(\boldsymbol{\varphi})L(\boldsymbol{\varphi})^T = C^{\mathrm{S}}_{\widetilde{\boldsymbol{u}},\widetilde{\boldsymbol{u}}}$. Consequently, our MCMC scheme simulates from the whitened marginal variational posterior,

$$q(\widetilde{\boldsymbol{\zeta}},\boldsymbol{\theta},\boldsymbol{\varphi}) \propto \mathrm{N}(\widetilde{\boldsymbol{\zeta}} \mid 0, I_M)\pi(\boldsymbol{\varphi})\pi(\boldsymbol{\theta})\sigma_\varepsilon^{-N}\exp\left(-\frac{1}{2\sigma_\varepsilon^2}\sum_{n=1}^{N}(y_n^2)-\frac{N\tau_z^2}{2\sigma_\varepsilon^2}\right)\times$$
$$\left[\exp\left(\frac{1}{2\sigma_\varepsilon^2}\sum_{i,j=1}^{M}\left((C^{\mathrm{NS}}_{\widetilde{\boldsymbol{z}},\widetilde{\boldsymbol{z}}})^{-1}\odot\boldsymbol{P}\right)_{ij}+\frac{1}{2\sigma_\varepsilon^4}\boldsymbol{y}^TB\left(C^{\mathrm{NS}}_{\widetilde{\boldsymbol{z}},\widetilde{\boldsymbol{z}}}+\sigma_\varepsilon^{-2}\boldsymbol{P}\right)^{-1}B^T\boldsymbol{y}\right)\frac{|C^{\mathrm{NS}}_{\widetilde{\boldsymbol{z}},\widetilde{\boldsymbol{z}}}|^{\frac{1}{2}}}{|C^{\mathrm{NS}}_{\widetilde{\boldsymbol{z}},\widetilde{\boldsymbol{z}}}+\sigma_\varepsilon^{-2}\boldsymbol{P}|^{\frac{1}{2}}}\right]_{\widetilde{\boldsymbol{u}}}, \tag{5.18}$$

where $\widetilde{\boldsymbol{u}}$ is replaced by $L(\boldsymbol{\varphi})\widetilde{\boldsymbol{\zeta}} + \boldsymbol{\mu}_u$ in the expression in brackets. Furthermore, when required, we can draw samples of $\widetilde{\boldsymbol{z}}$ from its conditional variational posterior:

$$\widetilde{\boldsymbol{z}} \mid \widetilde{\boldsymbol{u}},\boldsymbol{\theta},\boldsymbol{\varphi} \sim \mathrm{N}\left(\sigma_\varepsilon^{-2}C^{\mathrm{NS}}_{\widetilde{\boldsymbol{z}},\widetilde{\boldsymbol{z}}}\left(C^{\mathrm{NS}}_{\widetilde{\boldsymbol{z}},\widetilde{\boldsymbol{z}}}+\sigma_\varepsilon^{-2}\boldsymbol{P}\right)^{-1}B^T\boldsymbol{y}, C^{\mathrm{NS}}_{\widetilde{\boldsymbol{z}},\widetilde{\boldsymbol{z}}}\left(C^{\mathrm{NS}}_{\widetilde{\boldsymbol{z}},\widetilde{\boldsymbol{z}}}+\sigma_\varepsilon^{-2}\boldsymbol{P}\right)^{-1}C^{\mathrm{NS}}_{\widetilde{\boldsymbol{z}},\widetilde{\boldsymbol{z}}}\right). \tag{5.19}$$

5.3.1.3 Separable ARD case

The non-stationary separable ARD kernel shown in Eq. (5.6) is a special case of the non-stationary ARD kernel, where the correlation function is assumed to factorise across dimensions. Consequently, the derivations follow trivially from Section 5.3.1.1. However, it is important to note that in this case, the integrals required to compute the expected log likelihood also factorise across dimensions. More pre-

cisely, in this setting, the elements of $\boldsymbol{\beta}_n$ and P_n are

$$\begin{aligned} \beta_{n,m} &= \prod_{d=1}^{D} \mathbb{E}_{\pi(u_{nd}|\widetilde{\boldsymbol{u}}_{:d},\boldsymbol{\theta},\boldsymbol{\varphi})} \left[C^{\mathrm{NS},d}_{z_n,\widetilde{z}_m} \right], \\ P_{n,mm'} &= \prod_{d=1}^{D} \mathbb{E}_{\pi(u_{nd}|\widetilde{\boldsymbol{u}}_{:d},\boldsymbol{\theta},\boldsymbol{\varphi})} \left[C^{\mathrm{NS},d}_{z_n,\widetilde{z}_m} C^{\mathrm{NS},d}_{z_n,\widetilde{z}_{m'}} \right], \end{aligned} \tag{5.20}$$

where $C^{\mathrm{NS},d}$ denotes the d^{th} factor in the non-stationary separable ARD kernel. Thus, the intractable integrals are expectations with respect to univariate Gaussian random variables and again, Gaussian-Hermite quadrature can be employed. The optimal whitened variational posterior $q(\widetilde{\boldsymbol{\zeta}}_1, \ldots, \widetilde{\boldsymbol{\zeta}}_D, \boldsymbol{\theta}, \boldsymbol{\varphi})$ is given by Eq. (5.15), after making the appropriate substitutions for B and $\boldsymbol{P}$, using Eq. (5.20).

5.3.2 Overview of Gauss-Hermite quadrature

Evaluation of the expected log likelihood requires the computation of intractable integrals; namely, the elements of $\boldsymbol{\beta}_n$ and P_n for each data point $n = 1, \ldots, N$. Specifically, non-separable ARD kernels, require $N(M + M(M+1)/2)$ *multivariate* intractable integrals; isotropic kernels, require $N(M + M(M+1)/2)$ *univariate* intractable integrals; and separable ARD kernels, require $DN(M+M(M+1)/2)$ *univariate* intractable integrals. In the latter two cases when these quantities correspond to univariate integrals, one can employ Gauss-Hermite quadrature to obtain an approximation (in the general setting, more expensive multivariate Gauss-Hermite quadrature may be used). Here, we briefly review Gauss-Hermite quadrature.

For ease of exposition, let us suppose we are interested in computing

$$\mathbb{E}[f(v)] = \int f(v) \mathrm{N}(v \mid \mu_v, \sigma_v^2) dv = \frac{1}{\sqrt{\pi}} \int g(h) \exp(-h^2) dh,$$

where $g(h) = f(\sqrt{2}\sigma_v h + \mu_v)$. Gauss-Hermite quadrature of order J approximates this by

$$\mathbb{E}[f(v)] \approx \frac{1}{\sqrt{\pi}} \sum_{j=1}^{J} w_j g(h_j),$$

where h_j (for $j = 1, \ldots, J$) are the roots of the J^{th} order Hermite polynomial $H_J(h)$, and

$$w_j = \frac{\sqrt{\pi} 2^{J-1} J!}{\left(J H_{J-1}(h_j)\right)^2}.$$

The error of the quadrature approximation is given by

$$\varepsilon(f) = \frac{J!}{(2J)!2^J} g^{(2J)}(\xi),$$

for some ξ. Thus, Gauss-Hermite quadrature provides a good approximation if g is close to a polynomial of order $2J-1$. In practice, the weight and nodes are computed with numerical algorithms (e.g. Stroud and Secrest, 1966; Rutishauser, 1962).

5.3.3 Algorithms

We present algorithms to sample from the whitened marginal approximated posteriors (Eq. (5.15) and Eq. (5.18)). The samplers are detailed for the case when a squared exponential covariance function is employed for both the stationary and non-stationary processes, such that the ARD kernel reduces to a separable ARD kernel. Note however, that the algorithms can be easily adapted to other kernels.

To improve parameter identifiability, we make use of the empirical priors discussed in Chapter 3 to fix the magnitude and mean of the length-scale processes. Furthermore, we standarise the observations, $\boldsymbol{y}$, to have zero mean and unit variance, such that fixing $\tau_z^2 = 1$ is an appropriate assumption. In this case, the target distribution in Eq. (5.15) simplifies to

$$q(\widetilde{\boldsymbol{\zeta}}_1, \ldots, \widetilde{\boldsymbol{\zeta}}_D, \sigma_\varepsilon^2, \boldsymbol{\lambda}) \propto \left(\prod_{d=1}^{D} \mathrm{N}(\widetilde{\boldsymbol{\zeta}}_d \mid 0, I_M) \pi(\lambda_d) \right) \pi(\sigma_\varepsilon^2) \sigma_\varepsilon^{-N} \exp\left(-\frac{1}{2\sigma_\varepsilon^2} \sum_{n=1}^{N} (y_n^2) - \frac{N}{2\sigma_\varepsilon^2} \right)$$
$$\left[\exp\left(\frac{1}{2\sigma_\varepsilon^2} \sum_{i,j=1}^{M} \left((C^{\mathrm{NS}}_{\widetilde{\boldsymbol{z}},\widetilde{\boldsymbol{z}}})^{-1} \odot \boldsymbol{P} \right)_{ij} + \frac{1}{2\sigma_\varepsilon^4} \boldsymbol{y}^T B \left(C^{\mathrm{NS}}_{\widetilde{\boldsymbol{z}},\widetilde{\boldsymbol{z}}} + \sigma_\varepsilon^{-2} \boldsymbol{P} \right)^{-1} B^T \boldsymbol{y} \right) \frac{|C^{\mathrm{NS}}_{\widetilde{\boldsymbol{z}},\widetilde{\boldsymbol{z}}}|^{\frac{1}{2}}}{|C^{\mathrm{NS}}_{\widetilde{\boldsymbol{z}},\widetilde{\boldsymbol{z}}} + \sigma_\varepsilon^{-2} \boldsymbol{P}|^{\frac{1}{2}}} \right]_{\widetilde{U}},$$

where $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_D)$ denotes the length-scale parameters of the D second level GP priors. Similarly, the target in Eq. (5.18) reduces to

$$q(\widetilde{\boldsymbol{\zeta}}, \sigma_\varepsilon^2, \boldsymbol{\lambda}) \propto \mathrm{N}(\widetilde{\boldsymbol{\zeta}} \mid 0, I_M) \prod_{d=1}^{D} \pi(\lambda_d) \pi(\sigma_\varepsilon^2) \sigma_\varepsilon^{-N} \exp\left(-\frac{1}{2\sigma_\varepsilon^2} \sum_{n=1}^{N} (y_n^2) - \frac{N}{2\sigma_\varepsilon^2} \right)$$
$$\left[\exp\left(\frac{1}{2\sigma_\varepsilon^2} \sum_{i,j=1}^{M} \left((C^{\mathrm{NS}}_{\widetilde{\boldsymbol{z}},\widetilde{\boldsymbol{z}}})^{-1} \odot \boldsymbol{P} \right)_{ij} + \frac{1}{2\sigma_\varepsilon^4} \boldsymbol{y}^T B \left(C^{\mathrm{NS}}_{\widetilde{\boldsymbol{z}},\widetilde{\boldsymbol{z}}} + \sigma_\varepsilon^{-2} \boldsymbol{P} \right)^{-1} B^T \boldsymbol{y} \right) \frac{|C^{\mathrm{NS}}_{\widetilde{\boldsymbol{z}},\widetilde{\boldsymbol{z}}}|^{\frac{1}{2}}}{|C^{\mathrm{NS}}_{\widetilde{\boldsymbol{z}},\widetilde{\boldsymbol{z}}} + \sigma_\varepsilon^{-2} \boldsymbol{P}|^{\frac{1}{2}}} \right]_{\widetilde{\boldsymbol{u}}},$$

where $\boldsymbol{\lambda} = (\lambda_1 \ldots \lambda_D)$ denotes the length-scale parameters for the second level stationary ARD GP prior.

The proposed algorithms use a Metropolis-within-Gibbs style scheme, where

Algorithm 9 ARD Sparse Variational MCMC

Require: Target distribution: $q(\widetilde{\boldsymbol{\zeta}}_1, \ldots, \widetilde{\boldsymbol{\zeta}}_D, \sigma_\varepsilon^2, \lambda_1, \ldots, \lambda_D)$, initial states: $\widetilde{\boldsymbol{\zeta}}_1^{(0)}, \ldots, \widetilde{\boldsymbol{\zeta}}_D^{(0)}$, $\sigma_\varepsilon^{2^{(0)}}$, $\lambda_1^{(0)}, \ldots, \lambda_D^{(0)}$, iterations: T, quadrature order: J, node positions: $\boldsymbol{h} := \{h_j\}_{j=1}^J$, and weights: $\boldsymbol{w} := \{w_j\}_{j=1}^J$.

1: **for** $t = 1, \ldots, T$ **do**

2: Draw $\log(\sigma_\varepsilon^{2^{(t)}})$ using RW-MH($\log(\sigma_\varepsilon^{2^{(t-1)}})$, s_σ^2) ▷Alg. 2

3: Jointly sample $\widetilde{\boldsymbol{\zeta}}_1^{(t)}, \ldots, \widetilde{\boldsymbol{\zeta}}_D^{(t)}$ using ELL-SS($(\widetilde{\boldsymbol{\zeta}}_1^{(t-1)}, \ldots, \widetilde{\boldsymbol{\zeta}}_D^{(t-1)})$, $I_{M\times D}$) ▷ Alg. 4
Step 11 requires Q-ARD($J, \boldsymbol{h}, \boldsymbol{w}, \widetilde{\boldsymbol{\zeta}}_1^{(t)}, \ldots, \widetilde{\boldsymbol{\zeta}}_D^{(t)}, \lambda_1^{(t-1)}, \ldots, \lambda_D^{(t-1)}$)

4: Sample $\log(\lambda_1^{(t)})$ using RW-MH($\log(\lambda_1^{(t-1)})$, $s_{\lambda_1}^2$) ▷Alg 2
Step 3 requires Q-ARD($J, \boldsymbol{h}, \boldsymbol{w}, \widetilde{\boldsymbol{\zeta}}_1^{(t)}, \ldots, \widetilde{\boldsymbol{\zeta}}_D^{(t)}, \lambda_1^{(t)}, \lambda_2^{(t-1)}, \ldots, \lambda_D^{(t-1)}$)

5: $\cdots$

6: Sample $\log(\lambda_D^{(t)})$ using RW-MH($\log(\lambda_D^{(t)})$, $s_{\lambda_D}^2$) ▷Alg 2
Step 3 requires Q-ARD($J, \boldsymbol{h}, \boldsymbol{w}, \widetilde{\boldsymbol{\zeta}}_1^{(t)}, \ldots, \widetilde{\boldsymbol{\zeta}}_D^{(t)}, \lambda_1^{(t)}, \ldots, \lambda_D^{(t)}$)

7: Run adaptation for s_σ^2, $s_{\lambda_1}^2, \ldots, s_{\lambda_D}^2$

8: **end for**

9: **return** T samples from $q(\widetilde{\boldsymbol{\zeta}}_1, \ldots, \widetilde{\boldsymbol{\zeta}}_D, \sigma_\varepsilon^2, \lambda_1, \ldots \lambda_D)$

the whitened spatially varying parameters are sampled employing elliptical slice sampling (Ell-SS) (Murray et al., 2010) and the remaining parameters are drawn with an adaptive random walk Metropolis-Hastings (RW-MH) procedure (Roberts and Rosenthal, 2009, Section 3). Algorithm 9 describes the MCMC scheme employed for the separable ARD covariance function, which in turn employs Algorithm 10 to compute the quantities B and $\boldsymbol{P}$ through Gauss-Hermite quadrature.

Similarly, Algorithm 11 presents the sampler for the isotropic case described in Section 5.3.1.2. This algorithm employs the subroutine detailed in Algorithm 12 to approximate the relevant quantities in Eq. (5.17). Note that Algorithm 11 can also be easily adapted to the case of an isotropic GP prior on the log length-scale process.

For efficiency, the positions and associated weights required in the quadrature schemes of Algorithms 10 and 12 can be precomputed, prior to running the MCMC, and passed to the samplers. The R package fastGHQuad (Blocker, 2018) is employed to compute the weights and nodes.

The computational complexity of Algorithms 9 and 11 are $\mathcal{O}(DJNM^2+M^3)$ and $\mathcal{O}(JNM^2+M^3)$, respectively, per expected log-likelihood evaluation, in contrast to $\mathcal{O}(N^3)$ for a sampler over the true posterior. We highlight that the computations in Algorithm 10, steps 4-17, and in Algorithm 12, steps 5-12, can be done in parallel,

Algorithm 10 Gauss-Hermite quadratures for ARD case (Q-ARD)

Require: Quadrature order: J, node positions: $\boldsymbol{h} := \{h_j\}_{j=1}^J$, weights: $\boldsymbol{w} := \{w_j\}_{j=1}^J$, current states: $\widetilde{\boldsymbol{\zeta}}_1, \ldots, \widetilde{\boldsymbol{\zeta}}_D$, $\lambda_1, \ldots, \lambda_D$

procedure Q-ARD($J, \boldsymbol{h}, \boldsymbol{w}, \widetilde{\boldsymbol{\zeta}}_1, \ldots, \widetilde{\boldsymbol{\zeta}}_D, \lambda_1, \ldots, \lambda_D$)

  Set: $\{\boldsymbol{P}_{ij}\}_{i,j=1}^M \leftarrow 0$ and $\{B_{ij}\}_{i,j=1}^{N,M} \leftarrow 0$

  for $d = 1, \ldots, D$ **do**

    Set $\widetilde{\boldsymbol{u}}_d \leftarrow L_d \widetilde{\boldsymbol{\zeta}}_d + \boldsymbol{\mu}_{u_d}$

    Compute predictive mean and variance:

$$\boldsymbol{\mu}^*_{\boldsymbol{u}_{:d}} = \boldsymbol{\mu}_{\boldsymbol{u}_{:d}} + C^{\mathrm{S}}_{\boldsymbol{u}_{:d},\widetilde{\boldsymbol{u}}_{:d}} (C^{\mathrm{S}}_{\widetilde{\boldsymbol{u}}_{:d},\widetilde{\boldsymbol{u}}_{:d}})^{-1} (\widetilde{\boldsymbol{u}}_{:d} - \boldsymbol{\mu}_{u_d}),$$

$$\Omega^*_{\boldsymbol{u}_{:d}} = C^{\mathrm{S}}_{\boldsymbol{u}_{:d},\boldsymbol{u}_{:d}} - C^{\mathrm{S}}_{\boldsymbol{u}_{:d},\widetilde{\boldsymbol{u}}_{:d}} (C^{\mathrm{S}}_{\widetilde{\boldsymbol{u}}_{:d},\widetilde{\boldsymbol{u}}_{:d}})^{-1} C^{\mathrm{S}}_{\widetilde{\boldsymbol{u}}_{:d},\boldsymbol{u}_{:d}}$$

  end for

  for $n = 1, \ldots, N$ **do**

    for $d = 1, \ldots, D$ **do**

      for $j = 1 \ldots J$ **do**

        $u_{nd} \leftarrow \sqrt{2} \left(\Omega^*_{\boldsymbol{u}_d}\right)_{nn} h_j + \left(\boldsymbol{\mu}^*_{\boldsymbol{u}_d}\right)_n$

        $f_{jd} = w_j C^{\mathrm{NS},d}_{z_n,\widetilde{\boldsymbol{z}}}$ ▷ $C^{\mathrm{NS},d}_{z_n,\widetilde{\boldsymbol{z}}}$ depends on u_{nd} and $\widetilde{\boldsymbol{u}}$

        $g_{jd} = w_j \left[C^{\mathrm{NS},d}_{\widetilde{\boldsymbol{z}},z_n} C^{\mathrm{NS},d}_{z_n,\widetilde{\boldsymbol{z}}}\right]$ ▷ $C^{\mathrm{NS},d}_{z_n,\widetilde{\boldsymbol{z}}}$ depends on u_{nd} and $\widetilde{\boldsymbol{u}}$

      end for

      Approximate expectations:

$$\boldsymbol{\beta}_{nd} \approx \frac{1}{\sqrt{\pi}} \sum_{j=1}^J f_{jd}, \qquad P_{nd} \approx \frac{1}{\sqrt{\pi}} \sum_{j=1}^J g_{jd},$$

    end for

    Set $\boldsymbol{\beta}_n \leftarrow \prod_{d=1}^D \boldsymbol{\beta}_{nd}$ and $P_n \leftarrow \prod_{d=1}^D P_{nd}$

    $B_{n:} \leftarrow \boldsymbol{\beta}_n$ ▷ $B_{n:}$ denotes n-th row of B

    $\boldsymbol{P} \leftarrow \boldsymbol{P} + P_n$

  end for

  return B and $\boldsymbol{P}$

end procedure

leading to further computational gains.

5.4 Simulation study

We simulate $N = 1,000$ observations from the model described in Section 5.3.1.2 with domain $[0, 1]$, noise variance $\sigma_\varepsilon^2 = 0.02$, and stationary length-scale hyperpa-

Algorithm 11 Isotropic Sparse Variational MCMC

Require: Target distribution: $q(\widetilde{\boldsymbol{\zeta}}, \sigma_\varepsilon^2, \lambda_1, \ldots, \lambda_D)$, initial states: $\widetilde{\boldsymbol{\zeta}}^{(0)}$, $\sigma_\varepsilon^{2^{(0)}}$, $\lambda_1^{(0)}, \ldots, \lambda_D^{(0)}$, iterations: T, quadrature order: J, node positions: $\boldsymbol{h} := \{h_j\}_{j=1}^J$, and weights: $\boldsymbol{w} := \{w_j\}_{j=1}^J$.

1: **for** $t = 1, \ldots, T$ **do**

2: Draw $\log \sigma_\varepsilon^{2^{(t)}}$ using RW-MH($\log \sigma_\varepsilon^{2^{(t-1)}}$, s_σ^2) ▷Alg 2

3: Draw $\widetilde{\boldsymbol{\zeta}}^{(t)}$ using ELL-SS($\widetilde{\boldsymbol{\zeta}}^{(t-1)}$, I_M) ▷Alg. 4
Step 11 requires Q-I($J, \boldsymbol{h}, \boldsymbol{w}, \widetilde{\boldsymbol{\zeta}}^{(t)}, \lambda_1^{(t-1)}, \ldots, \lambda_D^{(t-1)}$)

4: Draw $\log \lambda_1^{(t)}$ using RW-MH($\log \lambda_1^{(t-1)}$, $s_{\lambda_1}^2$) ▷Alg 2
Step 3 requires Q-I($J, \boldsymbol{h}, \boldsymbol{w}, \widetilde{\boldsymbol{\zeta}}^{(t)}, \lambda_1^{(t)}, \lambda_2^{(t-1)}, \ldots, \lambda_D^{(t-1)}$)

5: $\cdots$

6: Draw $\log \lambda_D^{(t)}$ using RW-MH($\log \lambda_D^{(t)}$, $s_{\lambda_D}^2$) ▷Alg 2
Step 3 requires Q-I($J, \boldsymbol{h}, \boldsymbol{w}, \widetilde{\boldsymbol{\zeta}}^{(t)}, \lambda_1^{(t)}, \ldots, \lambda_D^{(t)}$)

7: Run adaptation for s_σ^2, $s_{\lambda_1}^2, \ldots, s_{\lambda_D}^2$

8: **end for**

9: **return** T samples from $q(\widetilde{\boldsymbol{\zeta}}, \sigma_\varepsilon^2, \lambda_1, \ldots, \lambda_D)$

rameter $\lambda = 0.1$. We run the proposed sampling scheme in Algorithm 11, employing different numbers of inducing points ($M = 30, 45, 60$) and different quadrature orders ($J = 4, 8, 10$) to understand better how sensitive the model is to these choices.

5.4.1 Selection of inducing points

A poor selection of the inducing points can lead to unsatisfactory posterior and predictive estimates. On one hand, a fast and computationally cheap strategy is to employ K-means clustering, but this is often less efficient than optimising the inducing locations (Hensman et al., 2015). On the other hand, treating the inducing points as variational parameters requires derivative calculations, which for our model are expensive and not straightforward to compute, and adds $M \times D$ parameters to the model. Consequently, we decide to use a strategy that is informed, yet simple to implement. The points are selected by maximising a lower bound to the sparse variational marginal likelihood of a stationary Gaussian process (which is available in closed form), following Titsias (2009). More precisely, we employ a stationary Gaussian process with squared exponential covariance function and optimise the inducing points using the GPstuff toolbox implemented in MATLAB (Vanhatalo

Algorithm 12 Gauss-Hermite quadratures for Isotopic case (Q-I)

Require: Quadrature order:J, node positions: $\boldsymbol{h} := \{h_j\}_{j=1}^J$, weights: $\boldsymbol{w} := \{w_j\}_{j=1}^J$, current states: $\widetilde{\boldsymbol{\zeta}}$, $\lambda_1, \ldots, \lambda_D$

procedure Q-I($J, \boldsymbol{h}, \boldsymbol{w}, \widetilde{\boldsymbol{\zeta}}, \lambda_1, \ldots, \lambda_D$)

  Set: $\{\boldsymbol{P}_{ij}\}_{i,j=1}^M \leftarrow 0$ and $\{B_{ij}\}_{i,j=1}^{N,M} \leftarrow 0$

  Set $\widetilde{\boldsymbol{u}} \leftarrow L_{\lambda_1,\ldots,\lambda_D}\widetilde{\boldsymbol{\zeta}} + \boldsymbol{\mu}_u$

  Compute predictive mean and variance:

$$\boldsymbol{\mu}_u^* = \boldsymbol{\mu}_{\boldsymbol{u}} + C^{\mathrm{S}}_{\boldsymbol{u},\widetilde{\boldsymbol{u}}}(C^{\mathrm{S}}_{\widetilde{\boldsymbol{u}},\widetilde{\boldsymbol{u}}})^{-1}(\widetilde{\boldsymbol{u}} - \boldsymbol{\mu}_u),$$

$$\Omega^*_{\boldsymbol{u}} = C^{\mathrm{S}}_{\boldsymbol{u},\boldsymbol{u}} - C^{\mathrm{S}}_{\boldsymbol{u},\widetilde{\boldsymbol{u}}}(C^{\mathrm{S}}_{\widetilde{\boldsymbol{u}},\widetilde{\boldsymbol{u}}})^{-1}C^{\mathrm{S}}_{\widetilde{\boldsymbol{u}},\boldsymbol{u}}$$

  for $n = 1, \ldots, N$ **do**

    for $j = 1 \ldots J$ **do**

      $u_n \leftarrow \sqrt{2}\left(\Omega^*_{\boldsymbol{u}}\right)_{nn} h_j + \left(\boldsymbol{\mu}^*_{\boldsymbol{u}}\right)_n$

      $f_j = w_j C^{\mathrm{NS}}_{z_n,\widetilde{\boldsymbol{z}}}$ ▷ $C^{\mathrm{NS}}_{z_n,\widetilde{\boldsymbol{z}}}$ depends on u_n and $\widetilde{\boldsymbol{u}}$

      $g_j = w_j \left[C^{\mathrm{NS}}_{\widetilde{\boldsymbol{z}},z_n} C^{\mathrm{NS}}_{z_n,\widetilde{\boldsymbol{z}}}\right]$ ▷ $C^{\mathrm{NS}}_{z_n,\widetilde{\boldsymbol{z}}}$ depends on u_n and $\widetilde{\boldsymbol{u}}$

    end for

    Approximate expectations:

$$\boldsymbol{\beta}_n \approx \frac{1}{\sqrt{\pi}}\sum_{j=1}^J f_j, \qquad P_n \approx \frac{1}{\sqrt{\pi}}\sum_{j=1}^J g_j$$

    $B_{n:} \leftarrow \boldsymbol{\beta}_n$ ▷$B_{n:}$ denotes n-th row of B

    $\boldsymbol{P} \leftarrow \boldsymbol{P} + P_n$

  end for

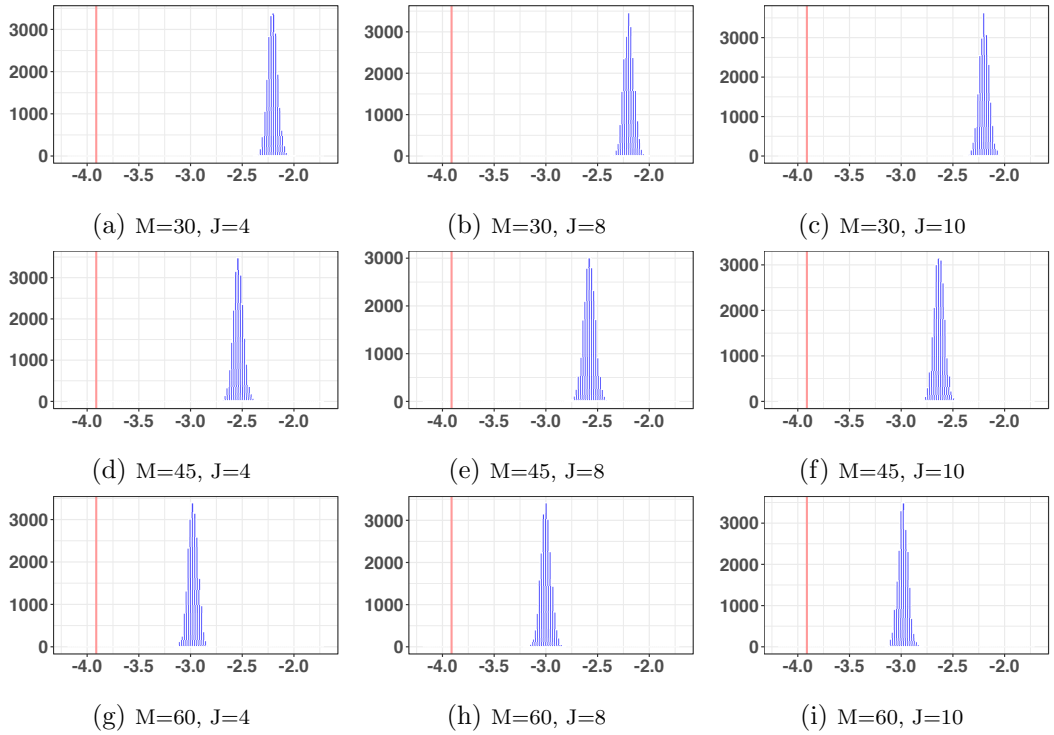
  return B and $\boldsymbol{P}$

end procedure

et al., 2015). The optimised inducing points are employed in the proposed MCMC scheme.

5.4.2 Posterior inference

We run the MCMC scheme described in Algorithm 11 for $T = 50,000$ iterations, employing the same initialisations across the varying number of inducing points $M = 30, 45, 60$ and quadrature orders $J = 4, 8, 10$. The first 30,000 iterations are discarded as burnin. First, the posterior of the log noise variance indicate that the parameter can be greatly over-estimated, especially when using few inducing points (see Figure 5.5). This can be the result of underestimation of the posterior



(a) M=30, J=4 (b) M=30, J=8 (c) M=30, J=10

(d) M=45, J=4 (e) M=45, J=8 (f) M=45, J=10

(g) M=60, J=4 (h) M=60, J=8 (i) M=60, J=10

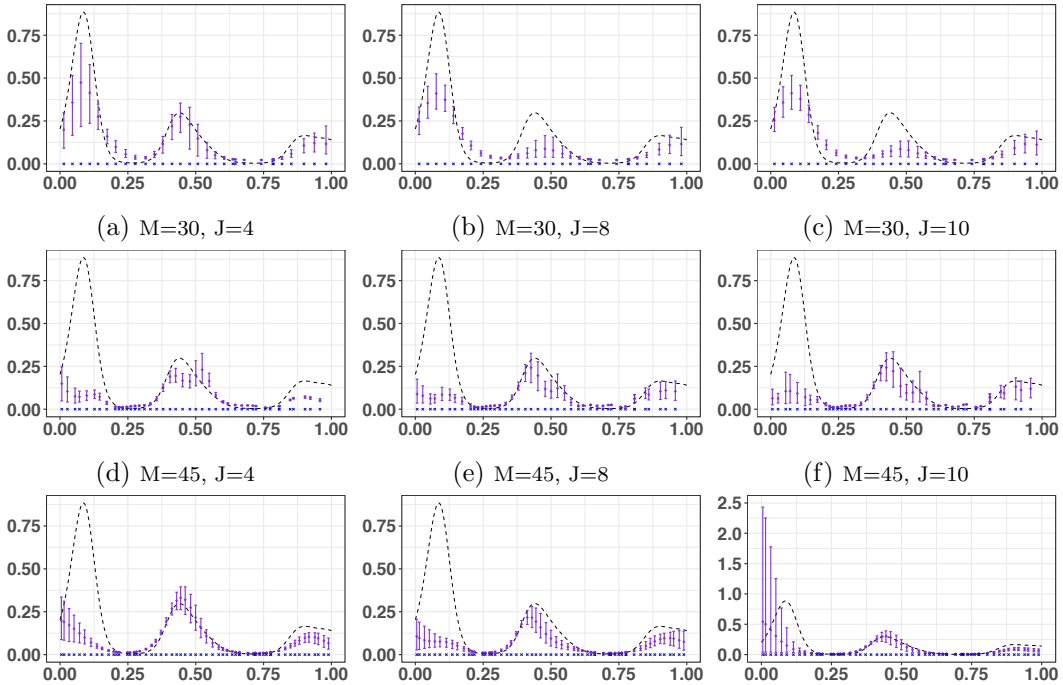
Figure 5.5: Histograms of the MCMC samples for the logarithm of the noise variance parameter for different numbers of inducing points ($M = 30, 45, 60$) and different quadrature orders ($J = 4, 8, 10$).

variance of the latent GP (and consequently, an overestimation of the noise variance to compensate (Gadd et al., 2018)). Second, posterior summaries of the length-scale process illustrated in Figure 5.6 highlight that increasing the number of inducing points and the number of nodes in the quadrature approximation does not necessarily result in more accurate posterior estimates.

While there is a clear computational benefit in employing the sparse variational posterior (see Table 5.1), there is also an important interplay between the number and locations of the inducing points and the order of the Gauss-Hermite quadrature approximation, that is further discussed in Section 5.4.4. The types of functions in the integrand vary considerably according to the distance between the observed locations x_n and each of the inducing points $\widetilde{x}_m$, as well as with respect to the values of the parameters in the covariance function.

	Full	$M = 30$			$M = 45$			$M = 60$		
	MCMC	$J = 4$	$J = 8$	$J = 10$	$J = 4$	$J = 8$	$J = 10$	$J = 4$	$J = 8$	$J = 10$
Avg. time (min)	15.13	1.17	1.17	0.83	1.44	1.60	1.32	4.09	6.51	8.18
Avg. evaluations	9.71	12.25	12.22	11.44	10.16	10.49	10.45	16.09	15.73	14.70

Table 5.1: Average time (in minutes) and likelihood evaluations required in the MCMC scheme. The average time required for 100 iterations is reported in minutes. The average number of likelihood evaluations in the elliptical slice sampler per iteration is reported.



(a) M=30, J=4 (b) M=30, J=8 (c) M=30, J=10

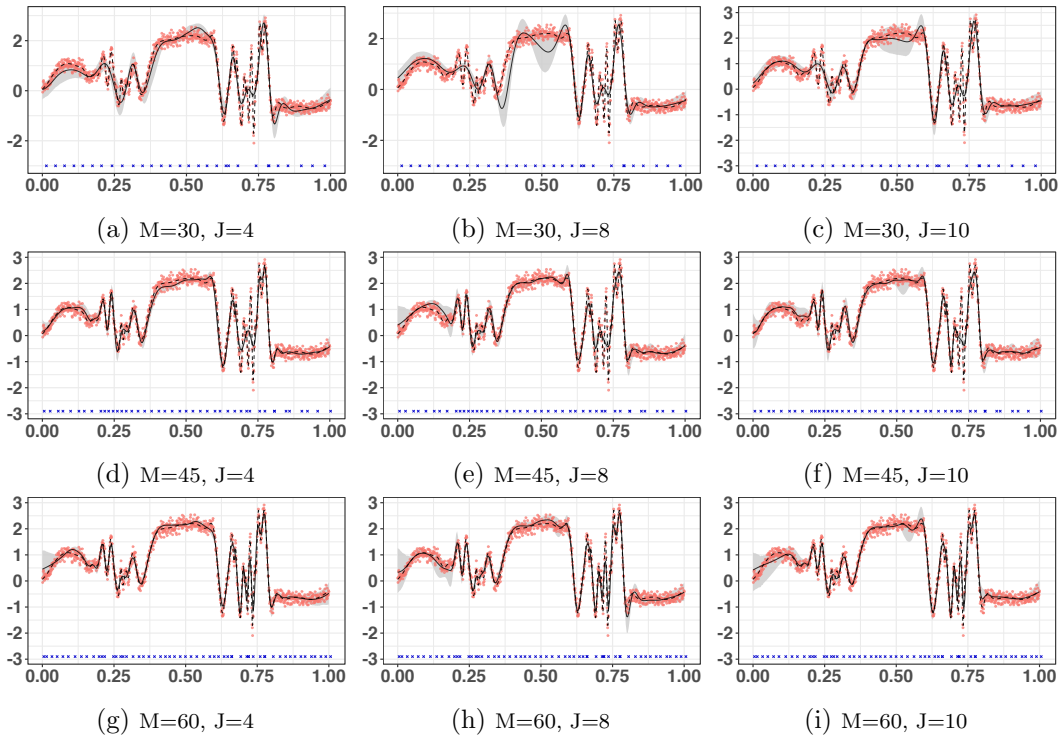
(d) M=45, J=4 (e) M=45, J=8 (f) M=45, J=10

(g) M=60, J=4 (h) M=60, J=8 (i) M=60, J=10

Figure 5.6: Spatially varying parameter $\ell(\cdot)$. The dashed line denotes the true process. Purple dots and bars show posterior estimates at the inducing locations with 95% HPD credible intervals.

5.4.3 Predictions

To evaluate the predictive performance of the method, we make out-of-sample predictions at 300 locations. Figure 5.7 shows predictive estimates of the non-stationary latent function. It is clear that when there are not enough inducing points or when they are not well located, important features of the function can be missed, see for instance Figures 5.7(a)-(c) in the range $[0.6, 0.8]$ and compared with Figures 5.7(g)-(i). Moreover, for a fixed number and location of the inducing points, the order of the quadrature approximation also affects the results. This is further



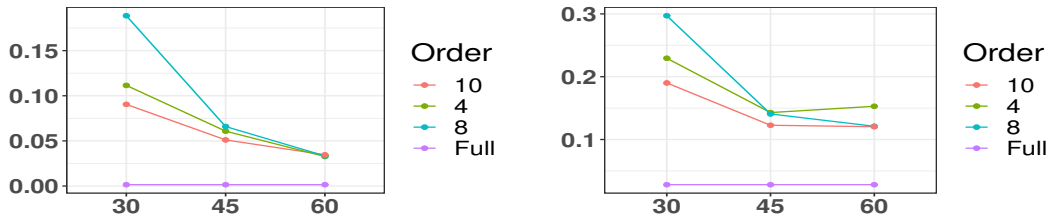
(a) M=30, J=4 (b) M=30, J=8 (c) M=30, J=10

(d) M=45, J=4 (e) M=45, J=8 (f) M=45, J=10

(g) M=60, J=4 (h) M=60, J=8 (i) M=60, J=10

Figure 5.7: Predictions for different numbers of inducing points ($M = 30, 45, 60$) and different quadrature orders ($J = 4, 8, 10$). The solid line denotes the predictive mean and the grey area depicts 95% HPD point-wise credible intervals. The dashed line denotes the true process.

emphasised in Figure 5.8, where, in some cases, the predictive error increases as we increase the number of nodes employed in the Gauss-Hermite approximation (see $J = 8$ for $M = 30$). Note also that with enough inducing points, e.g. $M = 60$ in this example, the quadrature order has less of an effect on the prediction errors.

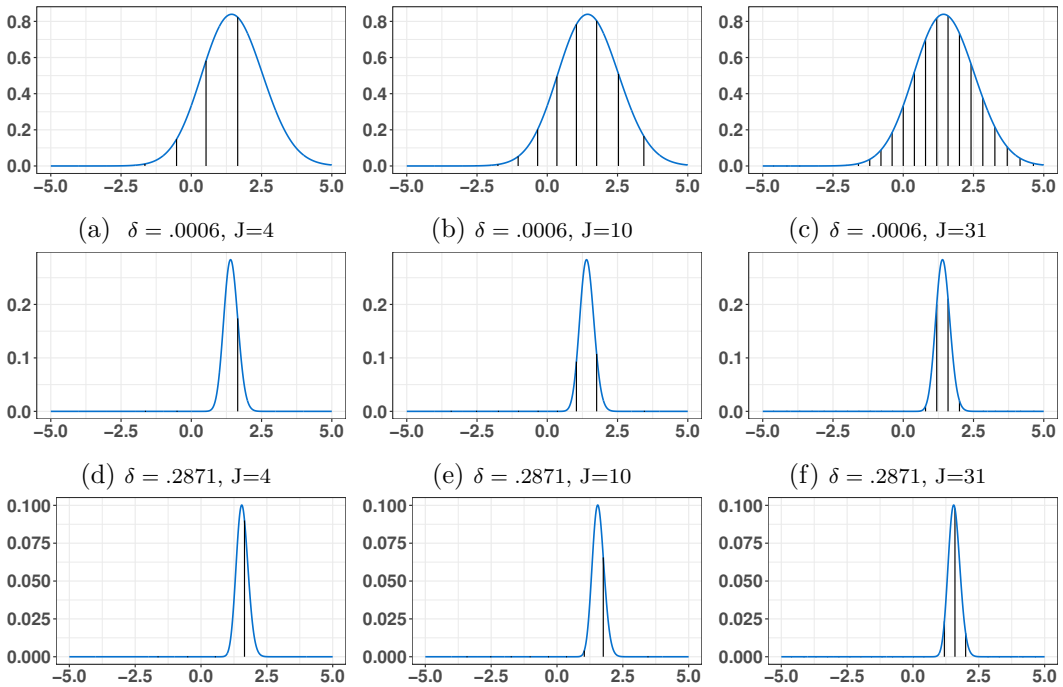


(a) MSE (b) MAE

Figure 5.8: MSE and MAE of the predictions for different numbers of inducing points ($M = 30, 45, 60$) and different quadrature orders ($J = 4, 8, 10$).

5.4.4 On the effect of the Gauss-Hermite quadrature

The implementation of the method requires Gauss-Hermite quadrature approximations of $\boldsymbol{\beta}_n$ and P_n, for each data point $n = 1, \ldots, N$. Here we analyse the integrand that corresponds to $\boldsymbol{\beta}_n$; however, similar behaviour will be encountered for P_n. For certain values of the hyperparameters and some values of $\delta = \|x_n - \widetilde{x}_m\|$, the integrand can have a sharp peak that is often located far from zero. This results in a poor approximation when Gauss-Hermite quadrature employs few nodes. Figure 5.9 illustrates how the shape of the integrand varies for different δ values (and a fixed hyperparameter value) and how the position of the nodes often misses the peak. Observe that as we increase δ, the function becomes sharper, and how fast this occurs is determined by the hyperparameters. Figure 5.9(f) illustrates clearly why the approximation can be poor, even with several nodes. We also highlight that in some cases, the node positions are located where the function is flat, see for instance Figure 5.9(h), which should depict ten horizontal bars but we can only observe two of them.



(a) $\delta = .0006$, J=4 (b) $\delta = .0006$, J=10 (c) $\delta = .0006$, J=31

(d) $\delta = .2871$, J=4 (e) $\delta = .2871$, J=10 (f) $\delta = .2871$, J=31

(g) $\delta = .9890$, J=4 (h) $\delta = .9890$, J=10 (i) $\delta = .9890$, J=31

Figure 5.9: The effect of the number of integrations points in Gauss-Hermite quadrature. The blue line denotes $g(h)\exp(-h^2)$ for a fixed hyperparameter value $\lambda = 0.1$. The vertical black lines depict the integrand evaluated at the node locations.

In summary, it appears that the optimal number of nodes varies according to the location of the inducing points and on the hyperparameter value. This is because the types of functions in the integrand vary considerably according to the distance between the observed location x_n and each of the inducing points $\widetilde{x}_m$, as well as with respect to the values of the parameters in the covariance function.

5.5 Discussion

We presented a novel variationally sparse formulation of 2-level Gaussian process models, which, combined with MCMC, results in an inference procedure of reduced computational complexity, compared to full MCMC, yet avoids the additional approximations and optimisation issues of variational inference. We derived three different optimal variational posterior distributions that correspond to different assumptions on the covariance structure. Finally, we outlined algorithms to sample from the derived posteriors.

As with any method that employs pseudo-inputs, its performance depends considerably on the choice of inducing locations. This dependence is more pronounced in non-stationary settings. Furthermore, approximating the intractable expectations through Gauss-Hermite quadrature adds to the computational complexity of the model. Our simulation studies suggest that the function we target cannot be approximated accurately with a small number of integration points, which would be ideal to keep computational cost at a minimum. The error of the quadrature approximations can be propagated to the estimates, resulting in poor inferences and consequently in poor predictive estimates. We anticipate that in higher dimensional non-stationary settings, these issues will only get worse. A promising route to tackle this is to employ a pseudo-marginal scheme (Beaumont, 2003; Andrieu et al., 2009). Thus, instead of approximating the needed expectations to compute the exponentiated expected log-likelihood, we use an unbiased estimator of it. We describe how this can be achieved in the following chapter.

CHAPTER 6

EXTENSIONS FOR VARIATIONALLY SPARSE MCMC MODELS

In Chapter 5, we derived free-form optimal sparse variational posterior distributions for 2-level Gaussian process (GP) models with different non-stationary kernels. Such optimal approximated posteriors require an exponentiated expected log-likelihood that is intractable. Moreover, accurately approximating the expectations required for its evaluation with Gauss-Hermite quadrature is computationally expensive. This chapter proposes an alternative approach to bypass this problem by replacing that costly likelihood evaluation with a computationally cheap estimate based on the block-Poisson estimator recently introduced by Quiroz et al. (2018). While this work is motivated by the 2-level Gaussian process regression (GPR) model, we emphasise that the method applies to any GP based model where the expected log-likelihood is not available in closed-form. Additionally, the scheme can make use of parallel computations to further speed up the inference procedure.

6.1 A signed block-Poisson pseudo-marginal scheme for variationally sparse 2-level GPs

The pseudo-marginal (PM) approach introduced by Beaumont (2003) and Andrieu et al. (2009) provides alternative routes to do exact Bayesian inference in models with intractable or expensive likelihoods. PM samplers employ a non-negative unbiased random estimator of the likelihood in place of the intractable function to produce samples from the exact posterior distribution. In particular, PM schemes have been previously employed to do inference in GP models (see, e.g. Filippone and Girolami, 2014; Murray and Graham, 2016; Xiong et al., 2017).

Recently, Quiroz et al. (2018) proposed combining an importance sampling sign correction (Lyne et al., 2015) with a product of Poisson estimators (Fearnhead et al., 2010) to derive a signed block-Poisson pseudo-marginal scheme for fast, exact inference in datasets with many observations. The approach of Quiroz et al. (2018) appears to be a promising direction towards the practical implementation of the non-stationary variationally sparse Markov chain Monte Carlo (MCMC) scheme introduced in Chapter 5. In the following, we outline how this can be achieved and present preliminary results. For simplicity, we focus on the isotropic case studied in Section 5.3.1.2; however, we highlight that the ideas here presented apply to both the non-separable and separable ARD models from Sections 5.3.1.1 and 5.3.1.3, respectively, as well as any variationally sparse GP-based model.

First, let us recall the optimal sparse variational distribution for the isotropic case:

$$q(\widetilde{\boldsymbol{z}}, \widetilde{\boldsymbol{u}}, \boldsymbol{\theta}, \boldsymbol{\varphi}) \propto \exp\left(\mathbb{E}_{\pi(\boldsymbol{z},\boldsymbol{u}|\widetilde{\boldsymbol{z}},\widetilde{\boldsymbol{u}},\boldsymbol{\theta},\boldsymbol{\varphi})}[\log(\mathrm{N}(\boldsymbol{y} \mid \boldsymbol{z}, \sigma_\varepsilon^2 I_N))]\right) \pi(\widetilde{\boldsymbol{z}} \mid \widetilde{\boldsymbol{u}}, \tau_z^2, \psi) \\ \pi(\widetilde{\boldsymbol{u}} \mid \boldsymbol{\varphi})\pi(\boldsymbol{\varphi})\pi(\boldsymbol{\theta}),$$

with $\boldsymbol{\theta} \coloneqq \{\sigma_\varepsilon^2, \tau_z^2, \psi\}$. Note that we do not employ the marginal posterior in Eq. (5.18) obtained by marginalising the function $\widetilde{\boldsymbol{z}}$ at the inducing locations. This is because we employ a variant of the PM scheme that employs a doubly stochastic estimator for the exponentiated expected log-likelihood, by also sub-sampling the data points. This provide computational gains, especially in larger settings; however, it requires us to write the log-likelihood as a sum over the observations, which is not possible for the marginal posterior $q(\widetilde{\boldsymbol{u}}, \boldsymbol{\theta}, \boldsymbol{\varphi})$. To break the correlation, we apply whitening to $\widetilde{\boldsymbol{z}}$ and $\widetilde{\boldsymbol{u}}$. Thus, we define the transformations: $\widetilde{\boldsymbol{z}} = L(\widetilde{\boldsymbol{u}})\widetilde{\boldsymbol{\xi}}$ and $\widetilde{\boldsymbol{u}} = L(\boldsymbol{\varphi})\widetilde{\boldsymbol{\zeta}} + \boldsymbol{\mu}_u$, where $\widetilde{\boldsymbol{\xi}} \sim \mathrm{N}(0, I_M)$, $\widetilde{\boldsymbol{\zeta}} \sim \mathrm{N}(0, I_M)$, $L(\widetilde{\boldsymbol{u}})L(\widetilde{\boldsymbol{u}})^T = C_{\boldsymbol{\phi}}^{\mathrm{NS}}$ and $L(\boldsymbol{\varphi})L(\boldsymbol{\varphi})^T = C_{\boldsymbol{\varphi}}^{\mathrm{S}}$. The whitened approximate variational posterior has the form

$$\begin{aligned}
&q(\widetilde{\boldsymbol{\xi}}, \widetilde{\boldsymbol{\zeta}}, \boldsymbol{\theta}, \boldsymbol{\varphi}) \\
&\propto \prod_{n=1}^{N} \exp\left(\mathbb{E}_{\pi(z_n,u_n|\widetilde{\boldsymbol{\xi}},\widetilde{\boldsymbol{\zeta}},\boldsymbol{\theta},\boldsymbol{\varphi})}[\log(\mathrm{N}(\boldsymbol{y} \mid \boldsymbol{z}, \sigma_\varepsilon^2 I_N))]\right) \mathrm{N}(\widetilde{\boldsymbol{\xi}} \mid 0, I_M)\pi(\boldsymbol{\theta})\mathrm{N}(\widetilde{\boldsymbol{\zeta}} \mid 0, I_M)\pi(\boldsymbol{\varphi}) \\
&\propto \prod_{n=1}^{N} \exp\left(-\frac{1}{2\sigma_\varepsilon^2}\left[\mathbb{E}_{\pi(u_n|\widetilde{\boldsymbol{\zeta}},\boldsymbol{\theta},\boldsymbol{\varphi})}\left((y_n - \mathbb{E}_{\pi(z_n|u_n,\widetilde{\boldsymbol{\xi}},\widetilde{\boldsymbol{\zeta}},\boldsymbol{\theta},\boldsymbol{\varphi})}[z_n])^2 + \mathbb{V}_{\pi(z_n|u_n,\widetilde{\boldsymbol{\xi}},\widetilde{\boldsymbol{\zeta}},\boldsymbol{\theta},\boldsymbol{\varphi})}[z_n]\right)\right]\right) \\
&\qquad \left(2\pi\sigma_\varepsilon^2\right)^{-\frac{N}{2}} \mathrm{N}(\widetilde{\boldsymbol{\xi}} \mid 0, I_M)\pi(\boldsymbol{\theta})\mathrm{N}(\widetilde{\boldsymbol{\zeta}} \mid 0, I_M)\pi(\boldsymbol{\varphi}),
\end{aligned}$$

where

$$\mathbb{E}_{\pi(z_n|u_n,\widetilde{\boldsymbol{\xi}},\widetilde{\boldsymbol{\zeta}},\boldsymbol{\theta},\boldsymbol{\varphi})}[z_n] = C^{\mathrm{NS}}_{z_n,\widetilde{\boldsymbol{z}}}(C^{\mathrm{NS}}_{\widetilde{\boldsymbol{z}},\widetilde{\boldsymbol{z}}})^{-1}L(\widetilde{\boldsymbol{u}})\widetilde{\boldsymbol{\xi}},$$
$$\mathbb{V}_{\pi(z_n|u_n,\widetilde{\boldsymbol{\xi}},\widetilde{\boldsymbol{\zeta}},\boldsymbol{\theta},\boldsymbol{\varphi})}[z_n] = \tau_z^2 - C^{\mathrm{NS}}_{z_n,\widetilde{\boldsymbol{z}}}(C^{\mathrm{NS}}_{\widetilde{\boldsymbol{z}},\widetilde{\boldsymbol{z}}})^{-1}C^{\mathrm{NS}}_{\widetilde{\boldsymbol{z}},z_n}.$$

Note that the dependence on $\widetilde{\boldsymbol{\zeta}}$ is through the non-stationary kernel that requires length-scales $\widetilde{\boldsymbol{\ell}} = \exp(\widetilde{\boldsymbol{u}})$.

Let us define:

$$l(y_n \mid \ell_n, \boldsymbol{\theta}, \boldsymbol{\varphi}, \widetilde{\boldsymbol{\xi}}, \widetilde{\boldsymbol{\zeta}}) = \frac{-1}{2\sigma_\varepsilon^2}\left[\left(y_n - C^{\mathrm{NS}}_{z_n,\widetilde{\boldsymbol{z}}}(C^{\mathrm{NS}}_{\widetilde{\boldsymbol{z}},\widetilde{\boldsymbol{z}}})^{-1}L(\widetilde{\boldsymbol{u}})\widetilde{\boldsymbol{\xi}}\right)^2 + \tau_z^2 - C^{\mathrm{NS}}_{z_n,\widetilde{\boldsymbol{z}}}(C^{\mathrm{NS}}_{\widetilde{\boldsymbol{z}},\widetilde{\boldsymbol{z}}})^{-1}C^{\mathrm{NS}}_{\widetilde{\boldsymbol{z}},z_n}\right] - \frac{1}{2}\log(2\pi\sigma_\varepsilon^2),$$

such that the distribution of interest can be written as

$$q(\widetilde{\boldsymbol{\xi}}, \widetilde{\boldsymbol{\zeta}}, \boldsymbol{\theta}, \boldsymbol{\varphi}) \propto \exp\left(\sum_{n=1}^{N} \mathbb{E}_{\pi(\ell_n|\widetilde{\boldsymbol{\zeta}},\boldsymbol{\theta},\boldsymbol{\varphi})}\left[l(y_n \mid \ell_n, \boldsymbol{\theta}, \boldsymbol{\varphi}, \widetilde{\boldsymbol{\xi}}, \widetilde{\boldsymbol{\zeta}})\right]\right) \mathrm{N}(\widetilde{\boldsymbol{\xi}} \mid 0, I_M)\pi(\boldsymbol{\theta}) \mathrm{N}(\widetilde{\boldsymbol{\zeta}} \mid 0, I_M)\pi(\boldsymbol{\varphi}), \tag{6.1}$$

where $\pi(\ell_n|\widetilde{\boldsymbol{\zeta}}, \boldsymbol{\theta}, \boldsymbol{\varphi}) = \text{Log-N}(c_n, w_n^2)$ with

$$\begin{aligned} c_n &= \mu_u + C^{\mathrm{S}}_{u_n,\widetilde{\boldsymbol{u}}}(C^{\mathrm{S}}_{\widetilde{\boldsymbol{u}},\widetilde{\boldsymbol{u}}})^{-1}(L(\boldsymbol{\varphi})\widetilde{\boldsymbol{\zeta}} + \boldsymbol{\mu}_u), \\ w_n^2 &= \tau_u^2 - C^{\mathrm{S}}_{u_n,\widetilde{\boldsymbol{u}}}(C^{\mathrm{S}}_{\widetilde{\boldsymbol{u}},\widetilde{\boldsymbol{u}}})^{-1}C^{\mathrm{S}}_{\widetilde{\boldsymbol{u}},u_n}. \end{aligned} \tag{6.2}$$

The exponentiated expected log-likelihood term in Eq. (6.1), which we define as

$$E = \exp\left(\sum_{n=1}^{N} \mathbb{E}_{\pi(\ell_n|\widetilde{\boldsymbol{\zeta}},\boldsymbol{\theta},\boldsymbol{\varphi})}\left[l(y_n \mid \ell_n, \boldsymbol{\theta}, \boldsymbol{\varphi}, \widetilde{\boldsymbol{\xi}}, \widetilde{\boldsymbol{\zeta}})\right]\right),$$

is intractable, and our goal is to find an unbiased estimator, $\widehat{E}$, of E, that is also computationally efficient for large sample sizes. To do so, we follow Quiroz et al. (2018) by employing subsampling for computationally efficiency and control variates to reduce the variance of our estimate. In addition, we employ an additional layer of stochasticity in our estimator to deal with the intractable expectation. The first step in this direction is to define the difference $d = \sum_{n=1}^{N} d_n$, with

$$d_n = \mathbb{E}_{\pi(\ell_n|\widetilde{\boldsymbol{\zeta}},\boldsymbol{\theta},\boldsymbol{\varphi})}\left[l(y_n \mid \ell_n, \boldsymbol{\theta}, \boldsymbol{\varphi}, \widetilde{\boldsymbol{\xi}}, \widetilde{\boldsymbol{\zeta}})\right] - \bar{\nu}_n, \tag{6.3}$$

where the control variate $\bar{\nu}_n$ is an approximation to $\mathbb{E}_{\pi(\ell_n|\widetilde{\boldsymbol{\zeta}},\boldsymbol{\theta},\boldsymbol{\varphi})}[l(y_n \mid \ell_n, \boldsymbol{\theta}, \boldsymbol{\varphi}, \widetilde{\boldsymbol{\xi}}, \widetilde{\boldsymbol{\zeta}})]$.

Thus, we have $E = d + \sum_{n=1}^{N} \bar{\nu}_n$. Specifically, we define $\bar{\nu}_n$ through a first-order Taylor-expansion around $\mathbb{E}[\ell_n]$:

$$\nu_n = l(y_n \mid \mathbb{E}[\ell_n], \boldsymbol{\theta}, \boldsymbol{\varphi}, \widetilde{\boldsymbol{\xi}}, \widetilde{\boldsymbol{\zeta}}) + (\ell_n - \mathbb{E}[\ell_n]) l'(y_n \mid \mathbb{E}[\ell_n], \boldsymbol{\theta}, \boldsymbol{\varphi}, \widetilde{\boldsymbol{\xi}}, \widetilde{\boldsymbol{\zeta}}), \tag{6.4}$$

such that

$$\bar{\nu}_n = \mathbb{E}_{\pi(\ell_n \mid \widetilde{\boldsymbol{\zeta}}, \boldsymbol{\varphi})}[\nu_n] = l(y_n \mid \mathbb{E}[\ell_n], \boldsymbol{\theta}, \boldsymbol{\varphi}, \widetilde{\boldsymbol{\xi}}, \widetilde{\boldsymbol{\zeta}}). \tag{6.5}$$

In this case, we can equivalently write the difference in Eq.(6.3) as

$$d_n = \mathbb{E}_{\pi(\ell_n \mid \widetilde{\boldsymbol{\zeta}}, \boldsymbol{\theta}, \boldsymbol{\varphi})} \left[l(y_n \mid \ell_n, \boldsymbol{\theta}, \boldsymbol{\varphi}, \widetilde{\boldsymbol{\xi}}, \widetilde{\boldsymbol{\zeta}}) - \nu_n \right],$$

and an unbiased difference estimator for d_n is

$$\widehat{d}_n = l(y_n \mid \ell_n, \boldsymbol{\theta}, \boldsymbol{\varphi}, \widetilde{\boldsymbol{\xi}}, \widetilde{\boldsymbol{\zeta}}) - \nu_n, \tag{6.6}$$

with $\ell_n \sim \pi(\ell_n \mid \widetilde{\boldsymbol{\zeta}}, \boldsymbol{\varphi})$. We then use subsampling techniques to obtain

$$\widehat{d}_B = \frac{N}{B} \sum_{b=1}^{B} \widehat{d}_{\alpha_b}, \tag{6.7}$$

where $\alpha_b \overset{\text{iid}}{\sim} \text{Unif}(1, \ldots, N)$ indexes a subsample of size B that is taken with replacement. The variance of the estimator in Eq. (6.7) is $\mathbb{V}[\widehat{d}_B] = \gamma / B$ with $\gamma = N^2 \mathbb{V}[\widehat{d}_{\alpha_b}]$ denoting the intrinsic variance of the estimator.

We re-write the difference estimator in Eq. (6.6) in terms of random variables that do not depend on $(\boldsymbol{\theta}, \boldsymbol{\varphi}, \widetilde{\boldsymbol{\xi}}, \widetilde{\boldsymbol{\zeta}})$. Thus, we introduce uniform random variables, $\chi_1, \ldots, \chi_N$, and apply the inverse cumulative distribution function (CDF) of a log-Normal with mean $\mu_{\ell_n} = \exp(c_n + w_n^2/2)$ and variance $\sigma_{\ell_n}^2 = \exp(2c_n + w_n^2)[\exp(w_n^2) - 1]$ to produce samples ℓ_n needed to evaluate Eq. (6.6).

Definition 6.1. The block-Poisson estimator is

$$\widehat{E} = \exp\left(\sum_{n=1}^{N} l(y_n \mid \mathbb{E}[\ell_n], \boldsymbol{\theta}, \boldsymbol{\varphi}, \widetilde{\boldsymbol{\xi}}, \widetilde{\boldsymbol{\zeta}}) \right) \prod_{k=1}^{\kappa} \exp\left(\frac{a + \kappa}{\kappa} \right) \prod_{h=1}^{\mathcal{H}_k} \left(\frac{\widehat{d}_B^{h,k} - a}{\kappa} \right) \tag{6.8}$$

with $\kappa \in \mathbb{Z}^+$, $\mathcal{H}_1, \ldots, \mathcal{H}_\kappa \sim \text{Pois}(1)$, $a \in \mathbb{R}$ a lower bound for

$$\widehat{d}_B^{h,k} = \frac{N}{B} \sum_{b=1}^{B} \widehat{d}_{\alpha_b}^{h,k}, \text{ with } \quad \widehat{d}_{\alpha_b}^{h,k} = l(y_{\alpha_b} \mid \chi_{\alpha_b}^{h,k}, \boldsymbol{\theta}, \boldsymbol{\varphi}, \widetilde{\boldsymbol{\xi}}, \widetilde{\boldsymbol{\zeta}}) - \nu_{\alpha_b}^{h,k},$$

where α_b is uniformly sampled from $(1, \ldots, N)$; $\chi_{\alpha_b}^{h,k} \sim \text{Unif}(0,1)$, and the the log-likelihood and Taylor expansion in $\widehat{d}_{\alpha_b}^{h,k}$ are evaluated with $\ell_{\alpha_b} = F_{\ell_n}^{-1}(\chi_{\alpha_b}^{h,k})$, where $F_{\ell_n}^{-1}$ denotes the inverse CDF of Log-N(c_n, w_n^2).

It is well known that efficient performance in PM algorithms is closely related to low variability in the estimators. Quiroz et al. (2018) assert that one of the key advantages of the estimator in Eq. (6.8) is that the blocking structure induce a controllable correlation between the estimates, which in turn reduces the variance of the estimators.

In order to employ the block-Poisson estimator, one must define a lower bound a for $\widehat{d}_B$. Quiroz et al. (2018, Section 3.3) explain that employing a soft lower bound, i.e. a lower bound that makes $\tau = \Pr(\widehat{E} \geq 0) \approx 1$, is computationally more efficient than employing a strict lower bound. To do so, the authors propose to work with the absolute value of $\widehat{E}$ to avoid possible negative likelihood estimates and then apply a sign correction to the estimates (Lyne et al., 2015). By minimising the variance of the estimator, the authors suggest such lower bound to be $a = \bar{d} - \kappa$, where $\bar{d}$ is an approximation of the difference d that can be calculated before the MCMC procedure. Also, they provide derivations to set the number of subsamples, B, and the number of Poisson estimators, κ, by minimising a measure of computational time (CT). This quantity describes the cost required to produce the equivalent of an independent Monte Carlo sample (Quiroz et al., 2018, see Section 4.3) and is derived under a normality assumption for $\widehat{d}_B$. The measure is defined to be:

$$\text{CT} = \kappa B \frac{\text{IF}}{(2\tau - 1)^2}, \tag{6.9}$$

with $\tau = \Pr(\widehat{E} \geq 0)$, and IF denoting an inefficiency measure of the MCMC samples; specifically

$$\text{IF} = 1 + 2\mathbb{E}_f\left(\frac{1-\vartheta}{\vartheta}\right), \quad \vartheta = \exp(-\beta + \omega^2/2)\Phi\left(\frac{\beta}{\omega} - \omega\right) + \Phi\left(\frac{-\beta}{\omega}\right),$$

with $\beta := f + \sigma^2_{\log|\widehat{E}|}$, $\omega := \sigma_{\log|\widehat{E}|}(1-\rho^2)^{1/2}$ and $f \sim \text{N}(\frac{1}{2}\sigma^2_{\log|\widehat{E}|}, \sigma^2_{\log|\widehat{E}|})$, where $\sigma^2_{\log|\widehat{E}|}$ denotes the variance of the logarithm of $|\widehat{E}|$ and ρ the induced correlation, which is controlled by the number of blocks. We remark that IF and τ are conditioned on values of the state, but to simplify notation we supress this dependency.

Note that the estimator in Eq. (6.8) is only unbiased without the absolute value.

Therefore, MCMC samples $(\widetilde{\boldsymbol{\xi}}, \widetilde{\boldsymbol{\zeta}}, \boldsymbol{\theta}, \boldsymbol{\varphi})$ do not follow $q(\widetilde{\boldsymbol{\xi}}, \widetilde{\boldsymbol{\zeta}}, \boldsymbol{\theta}, \boldsymbol{\varphi})$, but rather

$$\breve{q}(\widetilde{\boldsymbol{\xi}}, \widetilde{\boldsymbol{\zeta}}, \boldsymbol{\theta}, \boldsymbol{\varphi}) \propto \int |\widehat{E}| \pi(\widetilde{\boldsymbol{\xi}}, \widetilde{\boldsymbol{\zeta}}, \boldsymbol{\theta}, \boldsymbol{\varphi}) \pi(\mathcal{V}_1, \dots, \mathcal{V}_\kappa) \, d\mathcal{V}_1, \dots, \mathcal{V}_\kappa,$$

where $\pi(\widetilde{\boldsymbol{\xi}}, \widetilde{\boldsymbol{\zeta}}, \boldsymbol{\theta}, \boldsymbol{\varphi})$ is the joint prior over $(\widetilde{\boldsymbol{\xi}}, \widetilde{\boldsymbol{\zeta}}, \boldsymbol{\theta}, \boldsymbol{\varphi})$, $\pi(\mathcal{V}_1, \dots, \mathcal{V}_\kappa)$ the prior over the auxiliary random variates involved in the computation of the block-Poisson estimator with $\mathcal{V}_k = \{\mathcal{H}_k, \alpha_{1:B}^{1,k}, \dots, \alpha_{1:B}^{\mathcal{H}_k,k}, \chi_{1:B}^{1,k}, \dots, \chi_{1:B}^{\mathcal{H}_k,k}\}$ for $k = 1, \dots, \kappa$, where $\alpha_{1:B}^{h,k} \coloneqq (\alpha_1^{h,k}, \dots \alpha_B^{h,k})$ and $\chi_{1:B}^{h,k} \coloneqq (\chi_1^{h,k}, \dots \chi_B^{h,k})$. Nevertheless, one can estimate expectations with respect to $q(\widetilde{\boldsymbol{\xi}}, \widetilde{\boldsymbol{\zeta}}, \boldsymbol{\theta}, \boldsymbol{\varphi})$ by taking the MCMC samples from $\breve{q}(\widetilde{\boldsymbol{\xi}}, \widetilde{\boldsymbol{\zeta}}, \boldsymbol{\theta}, \boldsymbol{\varphi})$ and plugging them into an importance sampling step that corrects for the sign of the estimator (Lyne et al., 2015). For instance, assume one wants to compute the expectation of some function $g(\Psi)$ of some parameters Ψ on $(\widetilde{\boldsymbol{\xi}}, \widetilde{\boldsymbol{\zeta}}, \boldsymbol{\theta}, \boldsymbol{\varphi})$, then

$$\mathbb{E}_q[g(\Psi)] = \frac{\mathbb{E}_{\breve{q}}[g(\Psi)\mathcal{S}]}{\mathbb{E}_{\breve{q}}[\mathcal{S}]},$$

with $\mathcal{S} = \text{sign}(\widehat{E})$. Thus, one can estimate $\mathbb{E}_q[g(\Psi)]$ through

$$\widehat{\mathbb{E}}_q[g(\Psi)] = \frac{\sum_{t=1}^T g(\Psi^{(t)})\mathcal{S}^{(t)}}{\sum_{t=1}^T \mathcal{S}^{(t)}}.$$

Note that this requires to store the sign of $\widehat{E}$ at each iteration $t = 1, \dots, T$ of the MCMC proceedure.

In addition, while credible intervals reflecting uncertainty cannot be directly computed from the MCMC output, one can estimate the posterior that the parameters belong to a specified region, which in turn can be used to compute credible intervals. For example, consider the parameter σ_ε^2, letting $g(\Psi) = \mathbb{1}(\sigma_\epsilon^2 < s)$ for any s, the posterior CDF of σ_ε^2 is approximately:

$$\Pr(\sigma_\epsilon^2 < s \mid \mathcal{D}) = \mathbb{E}_q[\mathbb{1}(\sigma_\varepsilon^2 < s)] \approx \frac{\sum_{t=1}^T \mathcal{S}^{(t)} \mathbb{1}(\sigma_\varepsilon^2 < s)}{\sum_{t=1}^T \mathcal{S}^{(t)}}.$$

By evaluating this over a grid of s values, one can then compute credible intervals from the posterior CDF.

6.1.1 Algorithms

We present pseudo-code to sample from the whitened approximate posterior distribution for a variationally sparse 2-level GP model with an isotropic assumption for

the non-stationary kernel. Here, we fix some of the parameters employing empirical priors, obtaining the posterior:

$$q(\widetilde{\boldsymbol{\xi}}, \widetilde{\boldsymbol{\zeta}}, \sigma_\varepsilon^2, \boldsymbol{\lambda}) \propto \exp\left(\sum_{n=1}^{N} \mathbb{E}_{\pi(\ell_n|\widetilde{\boldsymbol{\zeta}}, \sigma_\varepsilon^2, \boldsymbol{\varphi})}\left[l(y_n \mid \ell_n, \sigma_\varepsilon^2, \boldsymbol{\lambda}, \widetilde{\boldsymbol{\xi}}, \widetilde{\boldsymbol{\zeta}})\right]\right) \mathrm{N}(\boldsymbol{\xi} \mid 0, I_M) \\ \pi(\sigma_\varepsilon^2)\mathrm{N}(\widetilde{\boldsymbol{\zeta}} \mid 0, I_M)\pi(\boldsymbol{\lambda}),$$

where $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_D)$.

Following the guidelines of Quiroz et al. (2018), we present in Algorithm 13 steps to derive the optimal tuning parameters needed for the signed block-Poisson pseudo-marginal (S-BP-PM) sampler. First, in order to determine an estimate of γ, and an approximation of the difference $\bar{d}$, we run a pilot MCMC to generate a small number of samples, S, from the posterior of interest. To do so, one can employ an MCMC sampler that employs $\bar{\nu}_n$ as a plug-in estimate of the intractable expectation in Eq. (6.1). In our case, we employ the already implemented MCMC algorithm based on Gauss-Hermite quadrature (see Chapter 5). Note also that any preferred MCMC scheme can be run on a small subset of the data to speed up computations. We employ a conservative estimate of γ that is the maximum of the estimated $\widehat{\gamma}^{(s)}$ across the initial MCMC samples. Second, for a fixed value of subsamples, the optimal κ is obtained based on a grid search to minimise a modified version of CT in Eq (6.9). Our CT measure also includes the number of inducing points as a penalisation and is given by

$$\mathrm{CT}^\star = \kappa B M^2 \frac{\mathrm{IF}}{(2\tau - 1)^2}.$$

In this case, this measure can also provide a guide to choosing the number of inducing points. Finally, to compute the difference estimator, we need to calculate ν_n (see Eq. (6.4)), which requires the derivative

$$l'(y_n \mid \mathbb{E}[\ell_n], \sigma_\varepsilon^2, \boldsymbol{\lambda}, \widetilde{\boldsymbol{\xi}}, \widetilde{\boldsymbol{\zeta}}) = \frac{1}{\sigma_\varepsilon^2}\left[\left(y_n - C^{\mathrm{NS}}_{z_n,\widetilde{\boldsymbol{z}}}(C^{\mathrm{NS}}_{\widetilde{\boldsymbol{z}},\widetilde{\boldsymbol{z}}})^{-1}L(\widetilde{\boldsymbol{u}})\widetilde{\boldsymbol{\xi}}\right)\frac{\partial C^{\mathrm{NS}}_{z_n,\widetilde{\boldsymbol{z}}}}{\partial \ell_n}(C^{\mathrm{NS}}_{\widetilde{\boldsymbol{z}},\widetilde{\boldsymbol{z}}})^{-1}L(\widetilde{\boldsymbol{u}})\widetilde{\boldsymbol{\xi}} + \\ \frac{\partial C^{\mathrm{NS}}_{z_n,\widetilde{\boldsymbol{z}}}}{\partial \ell_n}(C^{\mathrm{NS}}_{\widetilde{\boldsymbol{z}},\widetilde{\boldsymbol{z}}})^{-1}C^{\mathrm{NS}}_{\widetilde{\boldsymbol{z}},z_n}\right]$$

where the m^{th} entry of $\frac{\partial C^{\mathrm{NS}}_{z_n,\widetilde{\boldsymbol{z}}}}{\partial \ell_n}$ is given by $\frac{\partial C^{\mathrm{NS}}_{z_n,z_m}}{\partial \ell_n}$ with $C^{\mathrm{NS}}_{z_n,\widetilde{z}_m}$ obtained through Eq. (5.5). In this case, the derivative for the non-stationary isotropic squared expo-

nential (SE) kernel is:

$$\frac{\partial C^{\mathrm{NS}}_{z_n,\widetilde{z}_m}}{\partial \ell_n} = \left(\frac{\ell(\boldsymbol{x}_n)}{2\ell(\widetilde{\boldsymbol{x}}_m)} + \frac{\ell(\widetilde{\boldsymbol{x}}_m)}{2\ell(\boldsymbol{x}_n)}\right)^{-\frac{D}{2}} \exp\left(\frac{-\sum_{d=1}^{D}(x_{nd}-\widetilde{x}_{md})^2}{\ell^2(\boldsymbol{x}_n)+\ell^2(\widetilde{\boldsymbol{x}}_m)}\right) \times \left[\left(\frac{-D}{2\ell(\widetilde{\boldsymbol{x}}_m)} + \frac{D\ell(\widetilde{\boldsymbol{x}}_m)}{2\ell^2(\boldsymbol{x}_n)}\right) + \frac{2\ell(\boldsymbol{x}_n)}{(\ell^2(\boldsymbol{x}_n)+\ell^2(\widetilde{\boldsymbol{x}}_m))^2}\sum_{d=1}^{D}(x_{nd}-\widetilde{x}_{md})^2\right].$$

Algorithm 14 details the proposed MCMC sampler that employs the signed block-Poisson estimator. The scheme uses a Metropolis-within-Gibbs (MWG) sampler to iterate over the components of $\breve{q}(\widetilde{\boldsymbol{\xi}}, \widetilde{\boldsymbol{\zeta}}, \sigma_\varepsilon^2, \boldsymbol{\lambda})$. The noise variance σ_ε^2 and second level length-scale $\boldsymbol{\lambda}$, are sampled with adaptive random walk Metropolis-Hastings (RW-MH) steps. For the whitened spatially varying length-scale, we employ elliptical slice sampling (Ell-SS), and for the non-stationary function $\widetilde{\boldsymbol{z}}$, we use a Metropolis-Hastings (MH) step with a Gaussian proposal that approximates the true variational conditional posterior; more precisely, the proposal is:

$$\mathrm{N}\left(\sigma_\varepsilon^{-2} C^{\mathrm{NS}}_{\widetilde{\boldsymbol{z}},\widetilde{\boldsymbol{z}}}\left(C^{\mathrm{NS}}_{\widetilde{\boldsymbol{z}},\widetilde{\boldsymbol{z}}} + \sigma_\varepsilon^{-2}\widehat{\boldsymbol{P}}\right)^{-1}\widehat{B}^T\boldsymbol{y}, C^{\mathrm{NS}}_{\widetilde{\boldsymbol{z}},\widetilde{\boldsymbol{z}}}\left(C^{\mathrm{NS}}_{\widetilde{\boldsymbol{z}},\widetilde{\boldsymbol{z}}} + \sigma_\varepsilon^{-2}\widehat{\boldsymbol{P}}\right)^{-1} C^{\mathrm{NS}}_{\widetilde{\boldsymbol{z}},\widetilde{\boldsymbol{z}}}\right), \tag{6.10}$$

with $\widehat{B}$ an $N \times M$ matrix with rows $\widehat{\boldsymbol{\beta}}_n = \left[C^{\mathrm{NS}}_{z_n,\widetilde{\boldsymbol{z}}}\right]_{\mu_{\ell_n}}$ and $\widehat{\boldsymbol{P}} = \sum_{n=1}^{N}\widehat{P}_n$, with $\widehat{P}_n = \left[C^{\mathrm{NS}}_{\widetilde{\boldsymbol{z}},z_n} C^{\mathrm{NS}}_{z_n,\widetilde{\boldsymbol{z}}}\right]_{\mu_{\ell_n}}$, where we use $\mu_{\ell_n} = \exp(c_n + w_n^2/2)$ to evaluate the expressions in square brackets with c_n and w_n^2 as in Eq. (6.2).

The computational complexity of Algorithm 14 is $\mathcal{O}((\sum_{k=1}^{\kappa}\mathcal{H}_k)BM^2 + M^3 + NM))$. We emphasise that for all the parameter updates, we can compute the difference estimator $\widehat{d}^{h,g}_{\alpha_b}$, for all h and k, in parallel. While our current implementation does not make use of parallel computing, we adapt Algorithm 14 to vectorise some of the operations; for instance, c_n, w_n, $l(y_n \mid \ell_n, \sigma_\varepsilon^2, \boldsymbol{\lambda}, \widetilde{\boldsymbol{\xi}}, \widetilde{\boldsymbol{\zeta}})$, and $l'(y_n \mid \mathbb{E}[\ell_n], \sigma_\varepsilon^2, \boldsymbol{\lambda}, \widetilde{\boldsymbol{\xi}}, \widetilde{\boldsymbol{\zeta}})$ can be easily vectorised.

Finally, note that Algorithm 14 belongs to the so called grouped independence Metropolis–Hastings (GIMH) pseudo-marginal methods (Beaumont, 2003). While such algorithms are exact in the sense that its limiting distribution corresponds to the posterior of interest (Andrieu et al., 2009), it is known that they can suffer from slow mixing and convergence (Drovandi et al., 2018). In contrast, Monte Carlo within Metropolis (MCWM) schemes, which recompute the estimator at each iteration or step, tend to exhibit better mixing at the price of samples from an *approximate* posterior (Andrieu et al., 2009).

Algorithm 13 Optimal tuning parameters for S-BP-PM

Require: Target distribution: $q(\widetilde{\boldsymbol{\xi}}, \widetilde{\boldsymbol{\zeta}}, \sigma_\varepsilon^2, \boldsymbol{\lambda})$, small number of samples: S, subsample size: B', batch size B, a grid of κ values, assumption: $\widehat{d}_B^{h,k} \overset{\text{iid}}{\sim} \mathrm{N}(d, \gamma/B)$.

1: Employ Algorithm 11 with $J = 10$ to produce S samples from $q(\widetilde{\boldsymbol{\zeta}}, \sigma_\varepsilon^2, \boldsymbol{\lambda})$ based on a subsample.

2: Draw S samples from $\widetilde{\boldsymbol{z}} \mid \sigma_\varepsilon^2, \boldsymbol{\lambda}$ ▷See Eq. 5.19

3: **for** $s = 1, \dots S$ **do**

4: **for** $b = 1, \dots B'$ **do**

5: Sample $\alpha_b \sim \mathrm{Unif}(1, \dots N)$

6: Sample $\ell_{\alpha_b} \sim \text{Log-N}(c_{\alpha_b}, w_{\alpha_b}^2)$ where

$$c_{\alpha_b} = \mu_u + C^{\mathrm{S}}_{u_{\alpha_b}, \widetilde{\boldsymbol{u}}} (C^{\mathrm{S}}_{\widetilde{\boldsymbol{u}}, \widetilde{\boldsymbol{u}}})^{-1} (L(\boldsymbol{\lambda}^{(s)}) \widetilde{\boldsymbol{\zeta}}^{(s)} + \boldsymbol{\mu}_u),$$
$$w_{\alpha_b}^2 = \tau_u^2 - C^{\mathrm{S}}_{u_{\alpha_b}, \widetilde{\boldsymbol{u}}} (C^{\mathrm{S}}_{\widetilde{\boldsymbol{u}}, \widetilde{\boldsymbol{u}}})^{-1} C^{\mathrm{S}}_{\widetilde{\boldsymbol{u}}, u_{\alpha_b}}$$

7: Compute

$$\nu_{\alpha_b}^{(s)} = l(y_{\alpha_b} \mid \mathbb{E}[\ell_{\alpha_b}], \boldsymbol{\lambda}^{(s)}, \widetilde{\boldsymbol{\xi}}^{(s)}, \widetilde{\boldsymbol{\zeta}}^{(s)}) + (\ell_{\alpha_b} - \mathbb{E}[\ell_{\alpha_b}]) l'(y_{\alpha_b} \mid \mathbb{E}[\ell_{\alpha_b}], \boldsymbol{\lambda}^{(s)}, \widetilde{\boldsymbol{\xi}}^{(s)}, \widetilde{\boldsymbol{\zeta}}^{(s)})$$

8: Evaluate: $\widehat{d}_{\alpha_b}^{(s)} = l(y_{\alpha_b} \mid \ell_{\alpha_b}, \sigma_\varepsilon^{2(s)}, \boldsymbol{\lambda}^{(s)}, \widetilde{\boldsymbol{\xi}}^{(s)}, \widetilde{\boldsymbol{\zeta}}^{(s)}) - \nu_{\alpha_b}^{(s)}$

9: **end for**

10: Compute: $\widehat{\sigma^2}^{(s)} = \frac{1}{B'-1} \sum_{b=1}^{B'} \left(\widehat{d}_{\alpha_b}^{(s)} - \frac{1}{B'} \sum_{b=1}^{B'} \widehat{d}_{\alpha_b}^{(s)} \right)^2$

11: Compute: $\widehat{\gamma}^{(s)} = N^2 \widehat{\sigma^2}^{(s)}$

12: Compute $\widehat{d}_{B'}^{(s)} = \frac{N}{B'} \sum_{b=1}^{B'} \widehat{d}_{\alpha_b}^{(s)}$

13: **end for**

14: Set $\gamma_{max} = \max \widehat{\gamma}^{(s)}$

15: Compute $\bar{d} = \frac{1}{S} \sum_{s=1}^{S} \widehat{d}_{B'}^{(s)}$

16: **for each** κ **do**

17: Compute variance: $\sigma^2_{\log|\widehat{E}|} = \kappa(v^2 + \eta^2)$ with ▷Expectations approximated by truncation

$$v = \log\left(\sqrt{\frac{\gamma_{max}}{B\kappa^2}}\right) + \frac{1}{2}\left(\log 2 + \mathbb{E}_p[\psi^{(0)}(1/2 + p)]\right),$$
$$\eta^2 = \frac{1}{4}\left(\mathbb{E}_p[\psi^{(1)}(1/2 + p)] + \mathbb{V}_p[\psi^{(1)}(1/2 + p)]\right),$$

where $p \sim \mathrm{Pois}(B\kappa^2/2\gamma_{max})$, and $\psi^{(i)}$ the polygamma function of order i.

18: Compute the probability of a positive $\widehat{E}$: ▷Φ denotes CDF of a standard Gaussian

$$\tau = \frac{1}{2}\left(1 + \exp\left[2\kappa\left(\Phi\left(\frac{\kappa\sqrt{B}}{\sqrt{\gamma_{max}}}\right) - 1\right)\right]\right)$$

19: **if** $\kappa < 100$ **then** $\rho = 1 - \frac{1}{\kappa}$ **else** $\rho = 1 - \frac{1}{100}$ **end if**

20: Employ Gauss-Hermite quadrature to approximate: $\triangleright\ f \sim \mathrm{N}\left(\frac{1}{2}\sigma^2_{\log|\widehat{E}|}, \sigma^2_{\log|\widehat{E}|}\right)$

$$\widehat{\mathbb{E}}_f \approx \mathbb{E}_f\left(\frac{1-\vartheta}{\vartheta}\right), \quad \vartheta = \exp(-\beta + \omega^2/2)\Phi\left(\frac{\beta}{\omega} - \omega\right) + \Phi\left(\frac{-\beta}{\omega}\right),$$

with $\beta := f + \sigma^2_{\log|\widehat{E}|}$, $\omega = \sigma_{\log|\widehat{E}|}(1-\rho^2)^{1/2}$.

21: Compute the inefficiency: $\mathrm{IF} = 1 + 2\widehat{\mathbb{E}}_f$

22: Compute the computational time:

$$\mathrm{CT}^\star = \kappa B M^2 \frac{\mathrm{IF}}{(2\tau - 1)^2}$$

23: **end for**

24: **return** κ with the minimum $\mathrm{CT}^\star$ and $\bar{d}$.

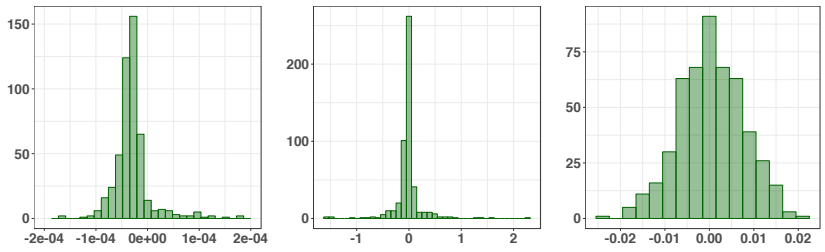
6.1.2 Preliminary results

Recall our simulation study of Section 5.4 in Chapter 5, where we use Gauss-Hermite quadrature to approximate the likelihood function. We aim to evaluate if the S-BP-PM scheme performs better than the quadrature approach in (i) recovering the true parameters, (ii) predictive performance, and (iii) computational time.

First, we attempt to determine the optimal tuning parameters for the S-BP-PM algorithm. To do so, we investigate the normality assumption of the estimator $\widehat{d}_{B=30}$ needed to employ Algorithm 13. This assumption is useful as it simplifies the computation of τ and IF in steps $17-18$ of Algorithm 13. Note that Quiroz et al. (2018) relax this by assuming the distribution of d_B is a mixture of normals. The algorithm however becomes much more complicated and the authors suggest instead to increase B until $\widehat{d}_B$ is approximately normal. Figure 6.1 shows histograms of 500 estimators for $M = 30, 45$, and 60. The normality assumption is confirmed for $M = 60$; however, both $M = 30$ and $M = 45$ appear to violate such assumption, as they depict heavy-tailed distributions. This suggests that we require a larger number of subsamples to ensure $\widehat{d}_B$ to be normally distributed. For both $M = 30$ and $M = 45$, we found that we necessitate at least $B = 300$ subsamples (see Figure E.1 and Figure E.2 in the Appendix) for normality to be a reasonable assumption.

To find the optimal number of Poisson estimators for a fixed value of B, we employ Algorithm 13. Figure 6.2, illustrates the results when $B = 30$ for the different values of M by depicting the logarithm of $\mathrm{CT}^\star$ as a function of κ for two different samples of the MCMC chains with five random subsamples each. The most conservative approach is to set the optimal value of κ at its maximum across different

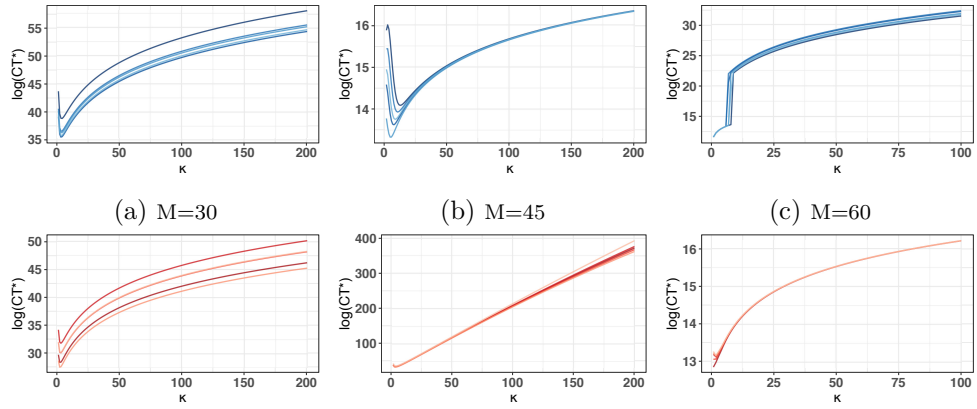
samples and subsamples. The results for the grid search with $B = 300$ indicate that for $M = 30$, $\kappa = 5$ (see (Figure E.3(a) and E.3(c)), which matches the obtained value when $B = 30$ with a significant increase in the CT^* measure. For $M = 45$, the optimal κ is found around 1800, attaining the maximum CT^* (Figure E.3(b) and E.3(d)). Because both CT^* and the overall computational complexity of Algorithm 14 depend on B as well as κ, one must be careful with setting such parameters to very large values, as this can undermine the computational gains of the method. In addition, Quiroz et al. (2018, Appendix S2) point out that when $\widehat{d}_B$ follows a Student t distribution, the optimal value of κ obtained under a normality assumption and $B = 30$ is comparable to that obtained when the estimator's distribution is approximated with a mixture of Gaussians (the guidelines under that assumption are not here investigated). Consequently, we employ $B = 30$ subsamples and we set the optimal κ values at 5, 13 and 2 for $M = 30, 45$, and 60, respectively (see Figure 6.2). Importantly, the minimum of $\mathrm{CT}^\star$ is attained for $M = 60$. Thus, an initial selection of the number of inducing points based on $\mathrm{CT}^\star$ is $M = 60$.



(a) M=30 (b) M=45 (c) M=60

Figure 6.1: Histograms of $\widehat{d}_B$ with $B = 30$ for different numbers of inducing points ($M = 30, 45, 60$).

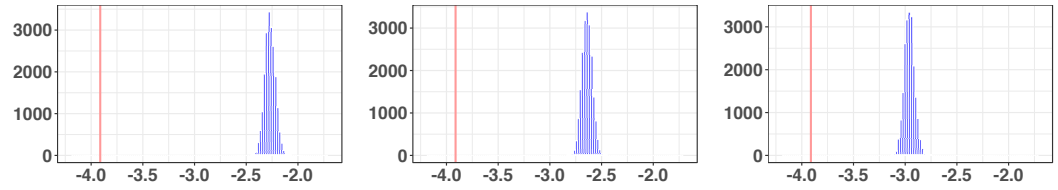
Second, to do inference, we use Algorithm 14. Because the sampler can have slow convergence, especially with poor initialisations, we employ an MCWM strategy for the first $5,000$ iterations and then discard the samples as part of the burnin phase. Note that this is not a necessary step, but our simulations suggest that employing an MCWM PM helps the sampler to move away faster from regions of low posterior probability. The resulting posterior estimates of the noise variance (Figure 6.3), are in accordance to those reported in Figure 5.5 (Section 5.4.2), showing that for the less inducing points, the noise variance is increasingly overestimated. For the length-scale process, we observe a systematic underestimation of the posterior variance (Figure 6.4). Despite whitening, the chains of some of the components of the log length-scale process show slow mixing (see Figure E.4).



(a) M=30 (b) M=45 (c) M=60

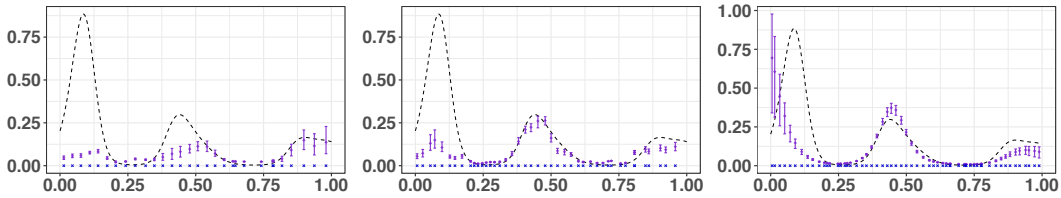
(d) M=30 (e) M=45 (f) M=60

Figure 6.2: Logarithm of the computational time measure $CT^{\star}$ with different values of κ for different number of inducing points M. Each row corresponds to a small number of samples ($S = 100$) from the pilot MCMC. Each line showcase the results of a random subsample. Optimal values of κ are set to 5, 13 and 2 for $M = 30, 45$, and 60 respectively.



(a) M=30 (b) M=45 (c) M=60

Figure 6.3: Histograms of the MCMC samples for the logarithm of the noise variance parameter for different numbers of inducing points ($M = 30, 45, 60$).



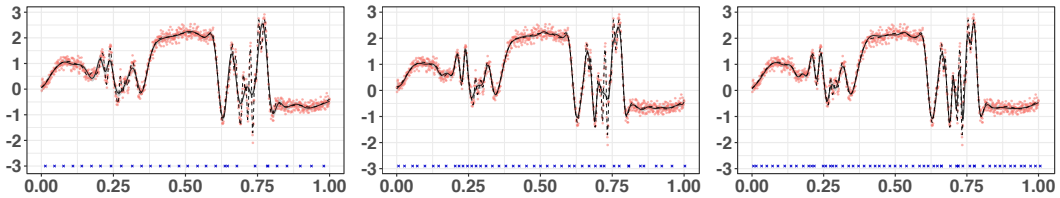
(a) M=30 (b) M=45 (c) M=60

Figure 6.4: Spatially varying parameter $\ell(\cdot)$. The dashed line denotes the true process. Purple dots and bars show posterior estimates at the inducing locations with 95% HPD credible intervals.

Figure 6.6(d) reports the computational time needed for inference with the S-BP-PM algorithm versus a full MCMC proceedure and the Gauss-Hermite quadrature

approach with different number of nodes (see Table E.1 for more details). In contrast to the performance of the three Gauss-Hermite approximations, the S-BP-PM scheme shows a reduction in time, when $M = 60$. This is in accordance to the results suggested by the CT* measure (see Figure 6.2), where we also observe that for $M = 45$ the computational time is at its maximum.

Finally, as in Section 5.4.3, we evaluate the predictive performance by making out-of-sample predictions at 300 locations. Figure 6.5 depicts the resulting predictive estimates of the latent function. While the function appears to be well recovered in regions with enough inducing points and indeed both the MAE and MSE are improved compared with the quadrature approach. However, we note that the uncertainty is greatly underestimated. More precisely, when comparing the attained



(a) M=30 (b) M=45 (c) M=60

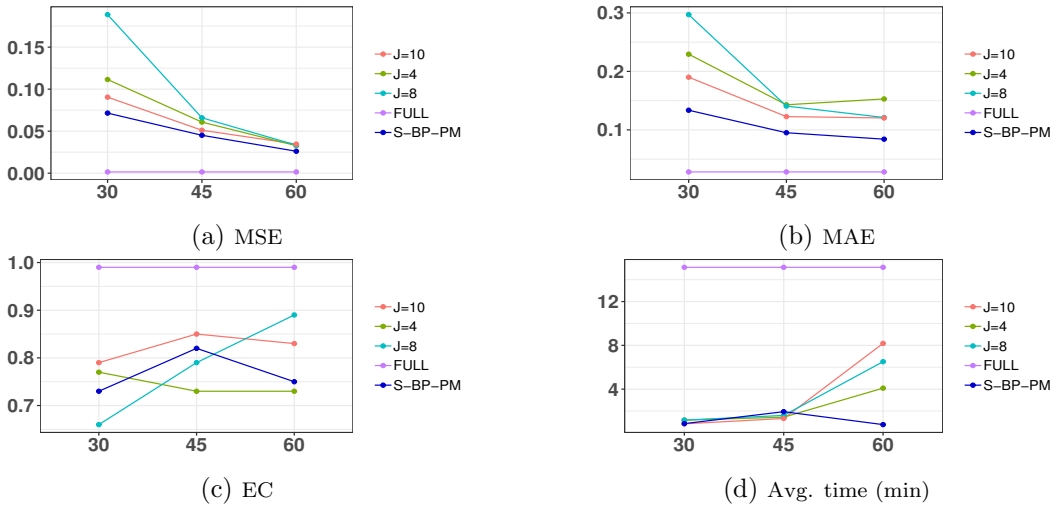
Figure 6.5: Predictions for different numbers of inducing points ($M = 30, 45, 60$). The solid line denotes the predictive mean and the grey area depicts 95% HPD point-wise credible intervals. The dashed line denotes the true process.

predictive performance with the Gauss-Hermite quadrature approach in the previous chapter, we observe a clear reduction in point-wise errors, with more significant differences for the MAE (see Figure 6.6(a)-(b)). While we expect an underestimation in the posterior variance inherited from the variational distribution, we note that there is a drop-off in EC compared to the results obtained when we use ten nodes for the quadrature approximation.

6.1.3 Discussion

Our preliminary results detailed in Section 6.1.2, shows that our proposed S-BP-PM scheme offers significant computational advantages, permitting to increase the number of inducing points (a crucial element to efficiently recover non-stationarities) without compromising scalability. Furthermore, the method appears to achieve better predictive performance in terms of MAE and MSE.

Importantly, the initial results suggest that the proposed modified CT measure can serve to select the number of inducing points during the search of the optimal



(a) MSE (b) MAE

(c) EC (d) Avg. time (min)

Figure 6.6: Comparison of the predictive performance and computational time. The results for S-BP-PM are shown in dark blue and compared with the Gauss-Hermite approximation approach with different number of nodes ($J = 4, 8, 10$) and the full (non-sparse) model. (a)-(c): Predictive performance comparison in terms of MSE, MAE and EC of out-of-sample predictions for different numbers of inducing points ($M = 30, 45, 60$). (d): Average computational time (in minutes) required for 100 MCMC iterations. The average considers the burnin phase.

tuning parameters. Also, it is essential to highlight that a considerable fraction of the inference time is spent in sampling the length-scale process; this is because each Ell-SS step needs several likelihood evaluations. We believe that replacing this step with a gradient informed MCMC sampler (e.g. Hamiltonian Monte Carlo (HMC)) could be a promising route to improve mixing and further speed up the inference procedure. Besides, automatic differentiation tools can be used to compute the derivatives needed for the difference estimator computation and for the sampler. This will permit the usage of the S-BP-PM method more generally, i.e. with other kernel matrices structures and in higher dimensions.

Note also, that the advantages offered by the S-BP-PM will be more evident in bigger datasets. Finally, we emphasise that the methodology is not specific to 2-level GP priors, see Section 6.2.

6.2 Further extensions

The signed block-Poisson pseudo-marginal (S-BP-PM) scheme from Section 6.1 can be seen as a more general inference framework that can be applied to several GP based statistical models. Specifically, for the (1-level) sparse variational MCMC

approach introduced by Hensman et al. (2015), one can employ the method whenever the expected log-likelihood is not available in closed form, avoiding N Gauss-Hermite quadrature approximations.

Recall the sparse variational posterior from Eq. (5.2)

$$q(\widetilde{\boldsymbol{z}}, \boldsymbol{\rho}, \boldsymbol{\phi}) \propto \exp\left(\sum_{n=1}^{N} \mathbb{E}_{\pi(z_n|\widetilde{\boldsymbol{z}},\boldsymbol{\phi})}[\log(p(y_n \mid z_n, \boldsymbol{\rho}))]\right) \pi(\widetilde{\boldsymbol{z}} \mid \boldsymbol{\phi})\pi(\boldsymbol{\phi})\pi(\boldsymbol{\rho}),$$

where $\pi(z_n \mid \widetilde{\boldsymbol{z}}, \boldsymbol{\rho}) = \mathrm{N}(z_n \mid \mu_z, \sigma_z^2)$. Now, let $l_n = \mathbb{E}_{\pi(z_n|\widetilde{\boldsymbol{z}},\boldsymbol{\phi})}[\log(p(y_n \mid z_n, \boldsymbol{\phi}))]$ and denote $\widehat{l_n}$ the Gauss-Hermite quadrature approximation

$$\widehat{l_n} = \frac{1}{\sqrt{\pi}} \sum_{j=1}^{J} w_j g(h_j),$$

with $g(h) = \log(p(y_n \mid \sqrt{2}\sigma_z h + \mu_z, \boldsymbol{\phi}))$, h_j denoting the roots of the J^{th} order Hermite polynomial $H_J(h)$, and $w_j = (\sqrt{\pi}2^{J-1}J!)/(JH_{J-1}(h_j))^2$. Further, let $l = \sum_{n=1}^{N} l_n$ and $\widehat{l} = \sum_{n=1}^{N} \widehat{l_n}$ and suppose

$$|l_n - \widehat{l_n}| \leq \epsilon.$$

Then,

$$|\exp(l) - \exp(\widehat{l})| = \exp(\min(l, \widehat{l}))(\exp(|l - \widehat{l}|) - 1) \leq \exp(\min(l, \widehat{l}))(\exp(N\epsilon) - 1).$$

Thus, there is the potential for the errors to become quite large, particularly if the expected log-likelihood is large or if the function in the integrand is very peaked and away from zero (as discussed in Section 5.4.4).

In the following, we enumerate some observation models where our proposed S-BP-PM method can be employed for faster and more accurate inference.

Classification with logistic link function: In binary classification problems, the response variable y_n corresponds to class labels taking values $0, 1$ with

$$p(y_n \mid z_n) = \left(\frac{\exp(z_n)}{1 + \exp(z_n)}\right)^{y_n} \left(\frac{1}{1 + \exp(z_n)}\right)^{1-y_n},$$
$$\log(p(y_n \mid z_n)) = y_n z_n - \log(1 + \exp(z_n)).$$

To compute l_n, consider a quadrature approximation to the second term with

$$g(h) = \log(1 + \exp(\sqrt{2}\sigma_z h + \mu_z)).$$

As the response is discrete, the likelihood will be bounded above by 1. However, the quadrature approximation error may not be sufficiently small, as the derivatives of g may not drop off, for example when $\sigma_z > 1$.

Ordered classification with logistic link function: Under this model, the response variable y_n takes ordered category values $c = 1, \ldots, C$ with cutoffs $\varepsilon_1 < \varepsilon_2 < \ldots < \varepsilon_{C-1}$. The cumulative probability of the response y_n with logit link is

$$\Pr(y_n \leq c \mid z, \boldsymbol{\varepsilon}) = \frac{\exp(\varepsilon_c - z)}{1 + \exp(\varepsilon_c - z)}.$$

The likelihood corresponds to

$$p(y_n = c \mid z_n, \boldsymbol{\varepsilon}) = \frac{\exp(\varepsilon_c - z_n)}{1 + \exp(\varepsilon_c - z_n)} - \frac{\exp(\varepsilon_{c-1} - z_n)}{1 + \exp(\varepsilon_{c-1} - z_n)},$$
$$\log(p(y_n = c \mid z_n, \boldsymbol{\varepsilon})) = \log\left(\frac{\exp(\varepsilon_c - z_n)}{1 + \exp(\varepsilon_c - z_n)} - \frac{\exp(\varepsilon_{c-1} - z_n)}{1 + \exp(\varepsilon_{c-1} - z_n)}\right).$$

Because the integral with respect to z_n is not analytically tractable, consider a Gauss-Hemite quadrature with

$$g(h) = \log\left(\frac{\exp(\varepsilon_c - \sqrt{2}\sigma_z h + \mu_z)}{1 + \exp(\varepsilon_c - \sqrt{2}\sigma_z h + \mu_z)} - \frac{\exp(\varepsilon_{c-1} - \sqrt{2}\sigma_z h + \mu_z)}{1 + \exp(\varepsilon_{c-1} - \sqrt{2}\sigma_z h + \mu_z)}\right).$$

Again, the derivatives of g may not decrease.

Multinomial classification with logistic link function: In a multi-class GP classifier, the response y_n takes class values $c = 0, \ldots, C$, with C denoting the total number of classes. In this model, the latent function $\boldsymbol{z}$ has dimension C, and for $c = 1, \cdots, C$,

$$p(y_n = c \mid \boldsymbol{z}) = \frac{\exp(z_{n,c})}{1 + \sum_{c=1}^{C} \exp(z_{n,c})}$$
$$\log(p(y_n = c \mid \boldsymbol{z})) = z_{n,c} - \log\left(1 + \sum_{c=1}^{C} \exp(z_{n,c})\right).$$

For $c = 0$,

$$p(y_n = 0 \mid \boldsymbol{z}) = \frac{1}{1 + \sum_{c=1}^{C} \exp(z_{n,c})}$$
$$\log(p(y_n = c \mid \boldsymbol{z})) = -\log\left(1 + \sum_{c=1}^{C} \exp(z_{n,c})\right).$$

In this case, the model requires from multivariate numerical integration; which can be computationally intensive, especially for large C.

GPR with student t likelihood: A GP regression framework that is robust to outliers (Neal, 1997) can be defined with the observation model:

$$p(y_n \mid z_n, \sigma^2, \nu) = \frac{\Gamma\left(\frac{\nu+1}{2}\right)}{\Gamma\left(\frac{\nu}{2}\right)\sqrt{\pi\nu\sigma^2}}\left(1 + \frac{1}{\nu}\frac{(y_n - z_n)^2}{\sigma^2}\right)^{-\frac{\nu+1}{2}},$$

where σ denotes the scale parameter and ν the degrees of freedom. The log-likelihood is then

$$\log(p(y_n \mid z_n)) = \log\left(\Gamma\left(\frac{\nu+1}{2}\right)\right) - \log\left(\Gamma\left(\frac{\nu}{2}\right)\right) - \frac{1}{2}\log(\pi\nu\sigma^2) - \frac{\nu+1}{2}\log\left(1 + \frac{1}{\nu}\frac{(y_n - z_n)^2}{\sigma^2}\right).$$

The model necessitates a quadrature approximation to the expectation of the final term with

$$g(h) = \log\left(1 + \frac{1}{\nu}\frac{(y - \sqrt{2}\sigma_z h - \mu_z)^2}{\sigma^2}\right).$$

This approximation may be poor in the case of small noise σ^2 with heavy tails, e.g. $\nu = 1$, and when σ_z is not too small.

Algorithm 14 Signed block-Poisson pseudo-marginal sampler (S-BP-PM)

Require: Target distribution: $\breve{q}(\widetilde{\boldsymbol{\xi}}, \widetilde{\boldsymbol{\zeta}}, \sigma_\varepsilon^2, \boldsymbol{\lambda})$, initial states: $\widetilde{\boldsymbol{\xi}}^{(0)}$,$\widetilde{\boldsymbol{\zeta}}^{(0)}$, ${\sigma_\varepsilon^2}^{(0)}$, $\lambda_1^{(0)}, \ldots, \lambda_D^{(0)}$, iterations: T, initial estimator $\widehat{E}^{(0)}$, number of groups: κ, batch size: B; initial means: $c_1^{(0)}, \ldots, c_n^{(0)}$, initial standard deviations: $w_1^{(0)}, \ldots, w_n^{(0)}$, groups $\mathcal{V}_1, \ldots \mathcal{V}_\kappa$, difference estimate: $\bar{d}$.

1: **for** $t = 1, \ldots, T$ **do**

2: Sample a block $g \sim \text{Unif}(1, \ldots, \kappa)$

3: Draw $\log {\sigma_\varepsilon^2}^{(t)}$ using RW-MH($\log {\sigma_\varepsilon^2}^{(t-1)}$, s_σ^2) ▷Alg. 2

Step 3 requires:

a. **for** $n = 1, \ldots, N$ **do** Compute $l_n = l(y_n | c_n^{(t-1)}, \sigma_\varepsilon^2, \boldsymbol{\lambda}^{(t-1)}, \widetilde{\boldsymbol{\xi}}^{(t-1)}, \widetilde{\boldsymbol{\zeta}}^{(t-1)})$ **end for**

b. Sample $\mathcal{H}_g' \sim \text{Pois}(1)$

for $h = 1, \ldots, \mathcal{H}_g'$ **do**

for $b = 1, \ldots, B$ **do**

Sample $\alpha_b^{h,g} \sim \text{Unif}(1, \ldots, N)$

Sample $\chi_b^{h,g} \sim \text{Unif}(0, 1)$

Compute $\ell_{\alpha_b}^{h,g} = F_{\ell_{\alpha_b}}^{-1}(\chi_b^{h,g})$ ▷ $\ell_{\alpha_b} \sim \text{Log-N}(c_{\alpha_b}^{(t-1)}, {w_{\alpha_b}^2}^{(t-1)})$

Evaluate:
$\nu_{\alpha_b}^{h,g} = l_{\alpha_b} + (\ell_{\alpha_b}^{h,g} - \mathbb{E}[\ell_{\alpha_b}]) l'(y_{\alpha_b} | \mathbb{E}[\ell_{\alpha_b}], \sigma_\varepsilon, \boldsymbol{\lambda}^{(t-1)}, \widetilde{\boldsymbol{\xi}}^{(t-1)}, \widetilde{\boldsymbol{\zeta}}^{(t-1)})$

Compute $\widehat{d}_{\alpha_b}^{h,g} = l(y_{\alpha_b} | \ell_{\alpha_b}^{h,g}, \sigma_\varepsilon^2, \boldsymbol{\lambda}^{(t-1)}, \widetilde{\boldsymbol{\xi}}^{(t-1)}, \widetilde{\boldsymbol{\zeta}}^{(t-1)}) - \nu_{\alpha_b}^{h,g}$

end for

Compute $\widehat{d}_B^{h,g} = \frac{N}{B} \sum_{b=1}^B \widehat{d}_{\alpha_b}^{h,g}$

end for

c. **for** $k \notin g$ **do**

for $h = 1, \ldots, \mathcal{H}_k$ **do**

for $b = 1, \ldots, B$ **do**

Evaluate:
$\nu_{\alpha_b}^{h,k} = l_{\alpha_b} + (\ell_{\alpha_b}^{h,k} - \mathbb{E}[\ell_{\alpha_b}]) l'(y_{\alpha_b} | \mathbb{E}[\ell_{\alpha_b}], \sigma_\varepsilon, \boldsymbol{\lambda}^{(t-1)}, \widetilde{\boldsymbol{\xi}}^{(t-1)}, \widetilde{\boldsymbol{\zeta}}^{(t-1)})$

Compute $\widehat{d}_{\alpha_b}^{h,k} = l(y_{\alpha_b} | \ell_{\alpha_b}^{h,k}, \sigma_\varepsilon^2, \boldsymbol{\lambda}^{(t-1)}, \widetilde{\boldsymbol{\xi}}^{(t-1)}, \widetilde{\boldsymbol{\zeta}}^{(t-1)}) - \nu_{\alpha_b}^{h,k}$

end for

Compute $\widehat{d}_B^{h,k} = \frac{N}{B} \sum_{b=1}^B \widehat{d}_{\alpha_b}^{h,k}$

end for

d. **end for**

e. Compute $\widehat{E}$

f. Store $\mathcal{S}_{\sigma_\varepsilon^2} = \text{sign}(\widehat{E})$

g. Evaluate: $\min\left\{1, \frac{|\widehat{E}| \pi\left(\log \sigma_\varepsilon^2\right)}{|\widehat{E}^{(t-1)}| \pi\left(\log {\sigma_\varepsilon^2}^{(t-1)}\right)}\right\}$

g. **if** $\log \sigma_\varepsilon^{2(t)}$ accepted **then** $\mathcal{H}_g \leftarrow \mathcal{H}'_g, \chi_{1:B}^{h,g^{(t-1)}} \leftarrow \chi_{1:B}^{h,g}, \alpha_{1:B}^{h,g^{(t-1)}} \leftarrow \alpha_{1:B}^{h,g}$, $h = 1, \dots, \mathcal{H}'_g$ $\widehat{E}^{(t)} \leftarrow \widehat{E}$ and $\mathcal{S}^{(t)}_{\sigma_\varepsilon^2} \leftarrow \mathcal{S}_{\sigma_\varepsilon^2}$ **end if**

4: Run Adaptation for s_1

5: Sample a block $g \sim \text{Unif}(1, \dots, \kappa)$

6: Compute Gaussian proposal p^* using Eq. 6.10

7: Draw $\widetilde{\boldsymbol{\xi}}^{(t)}$ using MH($\widetilde{\boldsymbol{\xi}}^{(t-1)}$, p^*) ▷Alg. 1

Step 3 requires:

a. **for** $n = 1, \dots, N$ **do** Compute $l_n = l(y_n | c_n^{(t-1)}, \sigma_\varepsilon^{2(t)}, \boldsymbol{\lambda}^{(t-1)}, \widetilde{\boldsymbol{\xi}}, \widetilde{\boldsymbol{\zeta}}^{(t-1)})$ **end for**

b. Sample $\mathcal{H}_g \sim \text{Pois}(1)$

for $h = 1, \dots, \mathcal{H}_g$ **do**

for $b = 1, \dots, B$ **do**

Sample $\alpha_b^{h,g} \sim \text{Unif}(1, \dots, N)$

Sample $\chi_b^{h,g} \sim \text{Unif}(0, 1)$

Compute $\ell_{\alpha_b}^{h,g} = F_{\ell_n}^{-1}(\chi_b^{h,g})$ ▷ $\ell_{\alpha_b} \sim \text{Log-N}(c_{\alpha_b}^{(t-1)}, w_{\alpha_b}^{2\ (t-1)})$

Evaluate:
$\nu_{\alpha_b}^{h,g} = l_{\alpha_b} + (\ell_{\alpha_b}^{h,g} - \mathbb{E}[\ell_{\alpha_b}]) l'(y_{\alpha_b} | \mathbb{E}[\ell_{\alpha_b}], \sigma_\varepsilon^{2(t)}, \boldsymbol{\lambda}^{(t-1)}, \widetilde{\boldsymbol{\xi}}, \widetilde{\boldsymbol{\zeta}}^{(t-1)})$

Compute $\widehat{d}_{\alpha_b}^{h,g} = l(y_{\alpha_b} | \ell_{\alpha_b}^{h,g}, \sigma_\varepsilon^{2(t)}, \boldsymbol{\lambda}^{(t-1)}, \widetilde{\boldsymbol{\xi}}, \widetilde{\boldsymbol{\zeta}}^{(t-1)}) - \nu_{\alpha_b}^{h,g}$

end for

Compute $\widehat{d}_B^{h,g} = \frac{N}{B} \sum_{b=1}^B \widehat{d}_{\alpha_b}^{h,g}$

end for

c. **for** $k \notin g$ **do**

for $h = 1, \dots, \mathcal{H}_k$ **do**

for $b = 1, \dots, B$ **do**

Evaluate:
$\nu_{\alpha_b}^{h,k} = l_{\alpha_b} + (\ell_{\alpha_b}^{h,k} - \mathbb{E}[\ell_{\alpha_b}]) l'(y_{\alpha_b} | \mathbb{E}[\ell_{\alpha_b}], \sigma_\varepsilon^{2(t)}, \boldsymbol{\lambda}^{(t-1)}, \widetilde{\boldsymbol{\xi}}, \widetilde{\boldsymbol{\zeta}}^{(t-1)})$

Compute $\widehat{d}_{\alpha_b}^{h,k} = l(y_{\alpha_b} \mid \ell_{\alpha_b}^{h,k}, \sigma_\varepsilon^{2(t)}, \boldsymbol{\lambda}^{(t-1)}, \widetilde{\boldsymbol{\xi}}, \widetilde{\boldsymbol{\zeta}}^{(t-1)}) - \nu_{\alpha_b}^{h,k}$

end for

Compute $\widehat{d}_B^{h,k} = \frac{N}{B} \sum_{b=1}^B \widehat{d}_{\alpha_b}^{h,k}$

end for

d. **end for**

e. Compute $\widehat{E}$

g. Store $\mathcal{S}_{\widetilde{\boldsymbol{\xi}}} = \text{sign}(\widehat{E})$

h. Evaluate: $\min\left\{1, \frac{|\widehat{E}| \text{N}\left(\widetilde{\boldsymbol{\xi}} | 0, I_M\right) p^*\left(\widetilde{\boldsymbol{z}}^{(t-1)}\right)}{|\widehat{E}^{(t-1)}| \text{N}\left(\widetilde{\boldsymbol{\xi}}^{(t-1)} | 0, I_M\right) p^*(\widetilde{\boldsymbol{z}})}\right\}$

i. Update $\chi_{1:B}^{h,g^{(t)}} \leftarrow \chi_{1:B}^{h,g}, \alpha_{1:B}^{h,g^{(t)}} \leftarrow \alpha_{1:B}^{h,g}$, for $h = 1, \dots, \mathcal{H}_g$

8: Sample a block $g \sim \text{Unif}(1, \dots, \kappa)$

9: Draw $\widetilde{\boldsymbol{\zeta}}^{(t)}$ using Ell-SS(I_M, $\widetilde{\boldsymbol{\zeta}}^{(t-1)}$) ▷Alg. 4

Step 3 requires:

a. **for** $n = 1, \dots N$ **do**

Update:

$$c_n^{(t)} = \mu_u + [C^{\mathrm{S}}_{u_n,\widetilde{\boldsymbol{u}}}(C^{\mathrm{S}}_{\widetilde{\boldsymbol{u}},\widetilde{\boldsymbol{u}}})^{-1}\widetilde{\boldsymbol{u}}]_{\widetilde{\boldsymbol{u}}=L(\boldsymbol{\lambda}^{(t-1)})\widetilde{\boldsymbol{\zeta}}+\boldsymbol{\mu}_u},$$

$$w_n^{2\,(t)} = \tau_u^2 - [C^{\mathrm{S}}_{u_n,\widetilde{\boldsymbol{u}}}(C^{\mathrm{S}}_{\widetilde{\boldsymbol{u}},\widetilde{\boldsymbol{u}}})^{-1}C^{\mathrm{S}}_{\widetilde{\boldsymbol{u}},u_n}]_{\widetilde{\boldsymbol{u}}=L(\boldsymbol{\lambda}^{(t-1)})\widetilde{\boldsymbol{\zeta}}+\boldsymbol{\mu}_u}$$

Compute $l_n = l(y_n|c_n^{(t)}, \sigma_\varepsilon^{2\,(t)}, \boldsymbol{\lambda}^{(t-1)}, \widetilde{\boldsymbol{\xi}}^{(t)}, \widetilde{\boldsymbol{\zeta}})$

b. **end for**

c. Sample $\mathcal{H}_g \sim$ Pois(1)

for $h = 1, \dots, \mathcal{H}_g$ **do**

for $b = 1, \dots, B$ **do**

Sample $\alpha_b^{h,g} \sim$ Unif$(1, \dots, N)$

Sample $\chi_b^{h,g} \sim$ Unif$(0, 1)$

Compute $\ell_{\alpha_b}^{h,g} = F_{\ell_n}^{-1}(\chi_b^{h,g})$ ▷ $\ell_{\alpha_b} \sim$ Log-N$(c_{\alpha_b}^{(t)}, w_{\alpha_b}^{2\ (t)})$

Evaluate:

$$\nu_{\alpha_b}^{h,g} = l_{\alpha_b} + (\ell_{\alpha_b}^{h,g} - \mathbb{E}[\ell_{\alpha_b}])l'(y_{\alpha_b}|\mathbb{E}[\ell_{\alpha_b}], \sigma_\varepsilon^{2\,(t)}, \boldsymbol{\lambda}^{(t-1)}, \widetilde{\boldsymbol{\xi}}^{(t)}, \widetilde{\boldsymbol{\zeta}})$$

Compute $\widehat{d}_{\alpha_b}^{h,g} = l(y_{\alpha_b}|\ell_{\alpha_b}^{h,g}, \sigma_\varepsilon^{2\,(t)}, \boldsymbol{\lambda}^{(t-1)}, \widetilde{\boldsymbol{\xi}}^{(t)}, \widetilde{\boldsymbol{\zeta}}) - \nu_{\alpha_b}^{h,g}$

end for

Compute $\widehat{d}_B^{h,k} = \frac{N}{B}\sum_{b=1}^B \widehat{d}_{\alpha_b}^{h,k}$

end for

d. **for** $k \notin g$ **do**

for $h = 1, \dots, \mathcal{H}_k$ **do**

for $b = 1, \dots, B$ **do**

Set $\ell_{\alpha_b}^{h,k} = F_{\ell_n}^{-1}(\chi_b^{h,k^{(t-1)}})$ ▷ $\ell_{\alpha_b} \sim$ Log-N$(c_{\alpha_b}^{(t)}, w_{\alpha_b}^{2\ (t)})$

Evaluate:

$$\nu_{\alpha_b}^{h,k} = l_{\alpha_b} + (\ell_{\alpha_b}^{h,k} - \mathbb{E}[\ell_{\alpha_b}])l'(y_{\alpha_b}|\mathbb{E}[\ell_{\alpha_b}], \sigma_\varepsilon^{2\,(t)}, \boldsymbol{\lambda}^{(t-1)}, \widetilde{\boldsymbol{\xi}}^{(t)}, \widetilde{\boldsymbol{\zeta}})$$

Compute $\widehat{d}_{\alpha_b}^{h,k} = l(y_{\alpha_b}|\ell_{\alpha_b}^{h,k}, \sigma_\varepsilon^{2\,(t)}, \boldsymbol{\lambda}^{(t-1)}, \widetilde{\boldsymbol{\xi}}^{(t)}, \widetilde{\boldsymbol{\zeta}}) - \nu_{\alpha_b}^{h,k}$

end for

Compute $\widehat{d}_B^{h,k} = \frac{N}{B}\sum_{b=1}^B \widehat{d}_{\alpha_b}^{h,k}$

end for

e. **end for**

f. Compute $|\widehat{E}^{(t)}|$

g. Store $\mathcal{S}_{\widetilde{\boldsymbol{\zeta}}}^{(t)} = \mathrm{sign}(\widehat{E}^{(t)})$

h. Update $\chi_{1:B}^{h,g\,(t)} \leftarrow \chi_{1:B}^{h,g}, \alpha_{1:B}^{h,g\,(t)} \leftarrow \alpha_{1:B}^{h,g}$, for $h = 1, \dots, \mathcal{H}_g$

10: Sample a block $g \sim$ Unif$(1, \dots, \kappa)$

11: Draw $\log \lambda_1^{(t)}$ using RW-MH($\log \lambda_1^{(t-1)}$, $s^2_{\lambda_1}$) ▷Alg. 2

Step 3 requires:

a. **for** $n = 1, \dots N$ **do**

Propose: $\triangleright \boldsymbol{\lambda} = (\lambda_1, \lambda_2^{(t-1)}, \dots, \lambda_D^{(t-1)})$

$$c_n = \mu_u + [C^{\mathrm{S}}_{u_n,\tilde{\boldsymbol{u}}}(C^{\mathrm{S}}_{\tilde{\boldsymbol{u}},\tilde{\boldsymbol{u}}})^{-1}\tilde{\boldsymbol{u}}]_{\tilde{\boldsymbol{u}}=L(\boldsymbol{\lambda})\tilde{\boldsymbol{\zeta}}^{(t)}+\boldsymbol{\mu}_u}, \text{ and}$$
$$w_n^2 = \tau_u^2 - [C^{\mathrm{S}}_{u_n,\tilde{\boldsymbol{u}}}(C^{\mathrm{S}}_{\tilde{\boldsymbol{u}},\tilde{\boldsymbol{u}}})^{-1}C^{\mathrm{S}}_{\tilde{\boldsymbol{u}},u_n}]_{\tilde{\boldsymbol{u}}=L(\boldsymbol{\lambda})\tilde{\boldsymbol{\zeta}}^{(t)}+\boldsymbol{\mu}_u}$$

Compute $l_n = l(y_n|c_n, \sigma_\varepsilon^{2^{(t)}}, \boldsymbol{\lambda}, \widetilde{\boldsymbol{\xi}}^{(t)}, \widetilde{\boldsymbol{\zeta}}^{(t)})$

b. **end for**

c. Sample $\mathcal{H}'_g \sim \mathrm{Pois}(1)$

for $h = 1, \dots, \mathcal{H}'_g$ **do**

for $b = 1, \dots, B$ **do**

Sample $\alpha_b^{h,g} \sim \mathrm{Unif}(1, \dots, N)$

Sample $\chi_b^{h,g} \sim \mathrm{Unif}(0, 1)$

Compute $\ell_{\alpha_b}^{h,g} = F^{-1}_{\ell_n}(\chi_b^{h,g})$ $\triangleright\ \ell_{\alpha_b} \sim \text{Log-N}(c_{\alpha_b}, w^2_{\alpha_b})$

Evaluate:

$$\nu_{\alpha_b}^{h,g} = l_{\alpha_b} + (\ell_{\alpha_b}^{h,g} - \mathbb{E}[\ell_{\alpha_b}])l'(y_{\alpha_b}|\mathbb{E}[\ell_{\alpha_b}], \sigma_\varepsilon^{2^{(t)}}, \boldsymbol{\lambda}, \widetilde{\boldsymbol{\xi}}^{(t)}, \widetilde{\boldsymbol{\zeta}}^{(t)})$$

Compute $\widehat{d}_{\alpha_b}^{h,g} = l(y_{\alpha_b}|\ell_{\alpha_b}^{h,g}, \sigma_\varepsilon^{2^{(t)}}, \boldsymbol{\lambda}, \widetilde{\boldsymbol{\xi}}^{(t)}, \widetilde{\boldsymbol{\zeta}}^{(t)}) - \nu_{\alpha_b}^{h,g}$

end for

Compute $\widehat{d}_B^{h,k} = \frac{N}{B}\sum_{b=1}^B \widehat{d}_{\alpha_b}^{h,k}$

end for

d. **for** $k \notin g$ **do**

for $h = 1, \dots, \mathcal{H}_k$ **do**

for $b = 1, \dots, B$ **do**

Set $\ell_{\alpha_b}^{h,k} = F^{-1}_{\ell_n}(\chi_b^{h,k^{(t)}})$ $\triangleright\ \ell_{\alpha_b} \sim \text{Log-N}(c_{\alpha_b}, w^2_{\alpha_b})$

Evaluate:

$$\nu_{\alpha_b}^{h,k} = l_{\alpha_b} + (\ell_{\alpha_b}^{h,k} - \mathbb{E}[\ell_{\alpha_b}])l'(y_{\alpha_b}|\mathbb{E}[\ell_{\alpha_b}], \sigma_\varepsilon^{2^{(t)}}, \boldsymbol{\lambda}, \widetilde{\boldsymbol{\xi}}^{(t)}, \widetilde{\boldsymbol{\zeta}}^{(t)})$$

Compute $\widehat{d}_{\alpha_b}^{h,k} = l(y_{\alpha_b}|\ell_{\alpha_b}^{h,k}, \sigma_\varepsilon^{2^{(t)}}, \boldsymbol{\lambda}, \widetilde{\boldsymbol{\xi}}^{(t)}, \widetilde{\boldsymbol{\zeta}}^{(t)}) - \nu_{\alpha_b}^{h,k}$

end for

Compute $\widehat{d}_B^{h,k} = \frac{N}{B}\sum_{b=1}^B \widehat{d}_{\alpha_b}^{h,k}$

end for

e. **end for**

f. Compute $\widehat{E}$

g. Store $\mathcal{S}_\lambda = \mathrm{sign}(\widehat{E})$

h. Evaluate: $\min\left\{1, \frac{|\widehat{E}|\pi(\log \lambda_1)}{|\widehat{E}^{(t)}|\pi\left(\log \lambda_1{}^{(t-1)}\right)}\right\}$

i. **if** $\log \lambda_1^{(t)}$ accepted **then** $c_n^{(t)} \leftarrow c_n$, $w_n^{(t)} \leftarrow w_n$ for all n, $\widehat{E}^{(t)} \leftarrow \widehat{E}$, $\mathcal{H}_g \leftarrow \mathcal{H}'_g$, $\mathcal{S}_\lambda^{(t)} \leftarrow \mathcal{S}_\lambda$, $\chi_{1:B}^{h,g^{(t)}} \leftarrow \chi_{1:B}^{h,g}, \alpha_{1:B}^{h,g^{(t)}} \leftarrow \alpha_{1:B}^{h,g}$, for $h = 1, \dots, \mathcal{H}_g$ **end if**

12: **end for**

CHAPTER 7

CONCLUSIONS

This section summarises the main findings and contributions of the work developed in this thesis and briefly discusses interesting extensions.

Multi-level Gaussian process (GP) priors provide a natural and meaningful approach to develop non-stationary models from a Bayesian perspective. Such models can be useful in numerous domains, where simpler approaches can fail to accurately represent important structures in the data; for instance, in environmental, geospatial and urban sciences, and computer emulation.

The heart of the models is a class of non-stationary covariance functions with input-varying length-scales that are modelled as random objects. Importantly, the proposed method can provide information about the correlation structure in the unknown. We note that it is possible to create similar hierarchies through other parameters in the covariance function; for instance, with the magnitude parameter. However, we advise against doing so with several parameters at a time as this will bring further issues of identifiability to the model and undermine interpretability. Besides, while this work has focused on non-stationary versions of the most common kernels; namely Matérn and squared exponential, other non-stationary covariances can be formulated in a similar fashion. For instance, periodic kernels that permit input-varying length-scales, magnitudes or even periods.

This thesis focused on Gaussian process regression (GPR) models, where conjugacy of the latent function permits marginalisation and results in a conditional posterior for the latent function that has an analytical solution. However, even when the latent function is marginalised, the posterior distribution is high-dimensional with elements that can be highly correlated and where some parameters can present identifiability issues. Thus, the model requires state-of-the-art computational tools

to devise practical inference algorithms.

Firstly, to improve parameter recovery, we suggested using the observed data to fix some of the parameters in an informed manner and to set weakly informative prior distributions for the length-scale process and the length-scale hyperprior.

Secondly, we used elliptical slice sampling (Ell-SS) as a central component of the algorithms developed. The choice of this algorithm is motivated due to its simple implementation and tuning-free nature, and because it does not need derivative information. Nevertheless, we acknowledge that other state-of-the-art sampling mechanisms, such as Hamiltonian Monte Carlo (HMC), can be effective and can be employed in place of Ell-SS. For derivative informed samplers, we recommend an implementation that uses automatic-differentiation tools rather than analytical solutions as for some kernels functions, derivatives with respect to the hyperparameters can be cumbersome and expensive to evaluate.

Finally, we have devised two strategies which bypass the computational bottlenecks of doing inference in 2-level GPR models. The first, presented in Chapter 4, is a sparse 2-level approach, that uses Markov chain Monte Carlo (MCMC) to sample from the exact posterior distribution with computational gains offered through a sparse representation of the inverse non-stationary covariance. The second, introduced in Chapter 5, derived a free-form low-dimensional approximation to the posterior, which is obtained by variational methods over an augmented distribution that employs inducing variables.

For the sparse 2-level GP model, we investigated and compared three different MCMC sampling algorithms. The best overall performance was attained with a marginal sampler; this algorithm is more expensive per iteration but results in chains with low autocorrelation. Besides, we concluded that an effective way to extend the model to D-dimensional settings is through an additive formulation that permits the choice between low-order or high-order interaction terms according to the problem under study. Key benefits of the additive approach are interpretability and scalability. Our experiments demonstrated the effectiveness of the method to recover several non-stationary features; more precisely, the model excelled in recovering edges, sharp peaks and also smooth variations. A crucial assumption of the method is that the observed data must lie on a grid; such a grid can be incomplete and non-equidistant. This type of data is common in geospatial problems and can be encountered from satellite measurements, image analysis or, as in our example, from computer emulators.

The variationally sparse framework for 2-level GPs developed low-dimensional approximated distributions under three different assumptions on the non-stationary

covariance function. The assumptions correpond to different constructions of the covariance function and input-varying parameters and were introduced to the model to control the number of parameters when $D \geq 2$. All three approximate posteriors have a likelihood term that is intractable as a consequence of integration over the inducing variables of the length-scale process. We designed MCMC algorithms to sample from the derived posteriors by approximating such expected log-likelihood with Gauss-Hermite quadrature; nevertheless, the types of functions we target appear not to be well approximated with few nodes, which is a key requirement of the model to control the computational overhead. Due to this constraint in the original formulation, Chapter 6 proposed to employ recent advances in the pseudo-marginal literature to avoid approximating the likelihood term. The developed sampling algorithm uses an unbiased but not necessarily positive estimator constructed as a product of Poisson estimators. Our empirical evaluation highlights that the employed pseudo-marginal algorithm improves predictive performance and reduces the computational cost, especially when we increase the number of inducing points. Adding enough inducing points to the model is vital to recover the signal in any model that employs this approach, but this becomes more critical for non-stationary datasets. Moreover, a significant advantage of this methodology is that it can make use of parallel computation to calculate the estimator, which can further speed up inference. Although we focused on 2-level GPR models, we emphasise that our inference framework could be successfully applied for single-level GPs with other (non-Gaussian) likelihood functions; for instance, Student-t.

We highlight that in contrast to the sparse 2-level construction of Chapter 4, this approach does not require the observed data to lie on a (possibly incomplete and non-equidistant) grid. However, we emphasise that a critical aspect in the performance of the variationally sparse MCMC approach is the selection of pseudo-input locations. This is a crucial step in any algorithm and inference framework that uses this type of sparsity to ameliorate the computational burden. Here we employed an informed but straightforward approach to set the inducing locations. A more advanced strategy can treat the locations as variational parameters, but it must be noticed that this will increase the number of parameters to be estimated and therefore, the computational complexity.

Finally, we remark that multi-level non-stationary GP priors can be used more broadly, i.e. in problems beyond standard regression. More precisely, we are currently investigating an extension of such non-stationary priors for problems that combine inversion and classification; such methods can be useful, for instance, in medical imaging.

In conclusion, this work provided new insights for efficient Bayesian inference in models with hierarchical GP priors that allow us to learn complex structures in the data common in several modern applications.

APPENDIX A

USEFUL IDENTITIES

Gaussian distribution

Joint distribution: Let us assume $\boldsymbol{a}$ is a random vector taking values in $\mathbb{R}^p$ and $\boldsymbol{b}$ is a random vector taking values in $\mathbb{R}^q$, and the joint distribution over $\boldsymbol{c} = (\boldsymbol{a}, \boldsymbol{b})^T$ is a $(p+q)$-dimensional Gaussian distribution with mean and variance,

$$\boldsymbol{m}_c = \begin{pmatrix} \boldsymbol{m}_a \\ \boldsymbol{m}_b \end{pmatrix} \quad \Sigma_c = \begin{pmatrix} \Sigma_a & \Sigma_{ab} \\ \Sigma_{ab}^T & \Sigma_b \end{pmatrix} \tag{A.1}$$

Marginal distribution: Assume

$$\boldsymbol{c} = \begin{pmatrix} \boldsymbol{a} \\ \boldsymbol{b} \end{pmatrix} \sim \mathrm{N}\left(\begin{pmatrix} \boldsymbol{m}_a \\ \boldsymbol{m}_b \end{pmatrix}, \begin{pmatrix} \Sigma_a & \Sigma_{ab} \\ \Sigma_{ab}^T & \Sigma_b \end{pmatrix} \right), \tag{A.2}$$

then the marginal distribution $\pi(\boldsymbol{a}) = \mathrm{N}(\boldsymbol{m}_a, \Sigma_a)$, and $\pi(\boldsymbol{b}) = \mathrm{N}(\boldsymbol{m}_b, \Sigma_b)$.

Conditional distribution: Assume

$$\boldsymbol{c} = \begin{pmatrix} \boldsymbol{a} \\ \boldsymbol{b} \end{pmatrix} \sim \mathrm{N}\left(\begin{pmatrix} \boldsymbol{m}_a \\ \boldsymbol{m}_b \end{pmatrix}, \begin{pmatrix} \Sigma_a & \Sigma_{ab} \\ \Sigma_{ab}^T & \Sigma_b \end{pmatrix} \right), \tag{A.3}$$

the conditional distribution $\pi(\boldsymbol{a} \mid \boldsymbol{b}) = \mathrm{N}\left(\boldsymbol{a} + \Sigma_{ab}\Sigma_b^{-1}(\boldsymbol{b} - \boldsymbol{m}_b), \Sigma_a + \Sigma_{ab}\Sigma_b^{-1}\Sigma_{ab}^T\right)$.

Product of two Gaussian distributions: The product of two Gaussian densities is proportional to a Gaussian density. Specifically

$$\mathrm{N}\left(\boldsymbol{a} \mid \boldsymbol{m}_{a_1}, \Sigma_{a_1}\right) \mathrm{N}\left(\boldsymbol{a} \mid \boldsymbol{m}_{a_2}, \Sigma_{a_2}\right) = \alpha \mathrm{N}(\boldsymbol{a} \mid \boldsymbol{m}_{a_3}, \Sigma_{a_3}),$$

with $\Sigma_{a_3} = \left(\Sigma_{a_1}^{-1} + \Sigma_{a_2}^{-1}\right)^{-1}$, $\boldsymbol{m}_{a_3} = \Sigma_{a_3}\left(\Sigma_{a_1}^{-1}\boldsymbol{m}_{a_1} + \Sigma_{a_2}^{-1}\boldsymbol{m}_{a_2}\right)$, and α the normalising constant. The normalising constant is also Gaussian both in $\boldsymbol{m}_{a_1}$ and in $\boldsymbol{m}_{a_2}$:

$$\det\left(2\pi(\Sigma_{a_1} + \Sigma_{a_2})\right)^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(\boldsymbol{m}_{a_1} - \boldsymbol{m}_{a_2})^T(\Sigma_{a_1} + \Sigma_{a_2})^{-1}(\boldsymbol{m}_{a_1} - \boldsymbol{m}_{a_2})\right).$$

Matrix inverse

Woodbury matrix identity: Asumme we have matrices $A \in \mathbb{R}^{p\times p}$, $U \in \mathbb{R}^{p\times q}$, $C \in \mathbb{R}^{q\times q}$ and $V \in \mathbb{R}^{q\times p}$. Then

$$(A + UCV)^{-1} = A^{-1} - A^{-1}U(C^{-1} + VA^{-1}U)^{-1}VA^{-1}.$$

Matrix determinant lemma: Asumme we have matrices $A \in \mathbb{R}^{p\times p}$, $U \in \mathbb{R}^{p\times q}$, $C \in \mathbb{R}^{q\times q}$ and $V \in \mathbb{R}^{q\times p}$. Then

$$\det\left(A + UCV^{\mathrm{T}}\right) = \det(C)\det(A)\det\left(C^{-1} + V^{\mathrm{T}}A^{-1}U\right).$$

APPENDIX B

SUPPLEMENTARY MATERIAL FOR CHAPTER 3

B.1 Synthetic data

SIM-1: $y(x) = \sin(x) + 2\exp(-30x^2) + \varepsilon, \quad$ where $\varepsilon \sim \mathrm{N}(0, 0.09)$.
SIM-2:

$$y(x) = \begin{cases} \sin(\frac{\pi x}{5}) + \frac{1}{5}\cos(\frac{4\pi x}{5}) + \varepsilon, & \text{if } x < 9.6 \\ \frac{x}{9} + \varepsilon, & \text{otherwise} \end{cases}, \quad \text{where } \varepsilon \sim \mathrm{N}(0, 0.01).$$

SIM-3:

$$y(x) = \sin(30(x-0.9)^4)\cos(2(x-0.9)) + .5(x-0.9) + \varepsilon, \quad \text{where } \varepsilon \sim \mathrm{N}(0, 0.0042)$$

SIM-4:

$$y(\boldsymbol{x}) = \frac{2}{\sqrt{3c}\pi^{1/4}}\left(1 - \frac{x_1^2 + x_2^2}{c^2}\right)\exp\left(\frac{-x_1^2 - x_2^2}{2c^2}\right) + \varepsilon,$$

where $c = 1.1$ and $\varepsilon \sim \mathrm{N}(0, 0.008)$.
SIM-5:

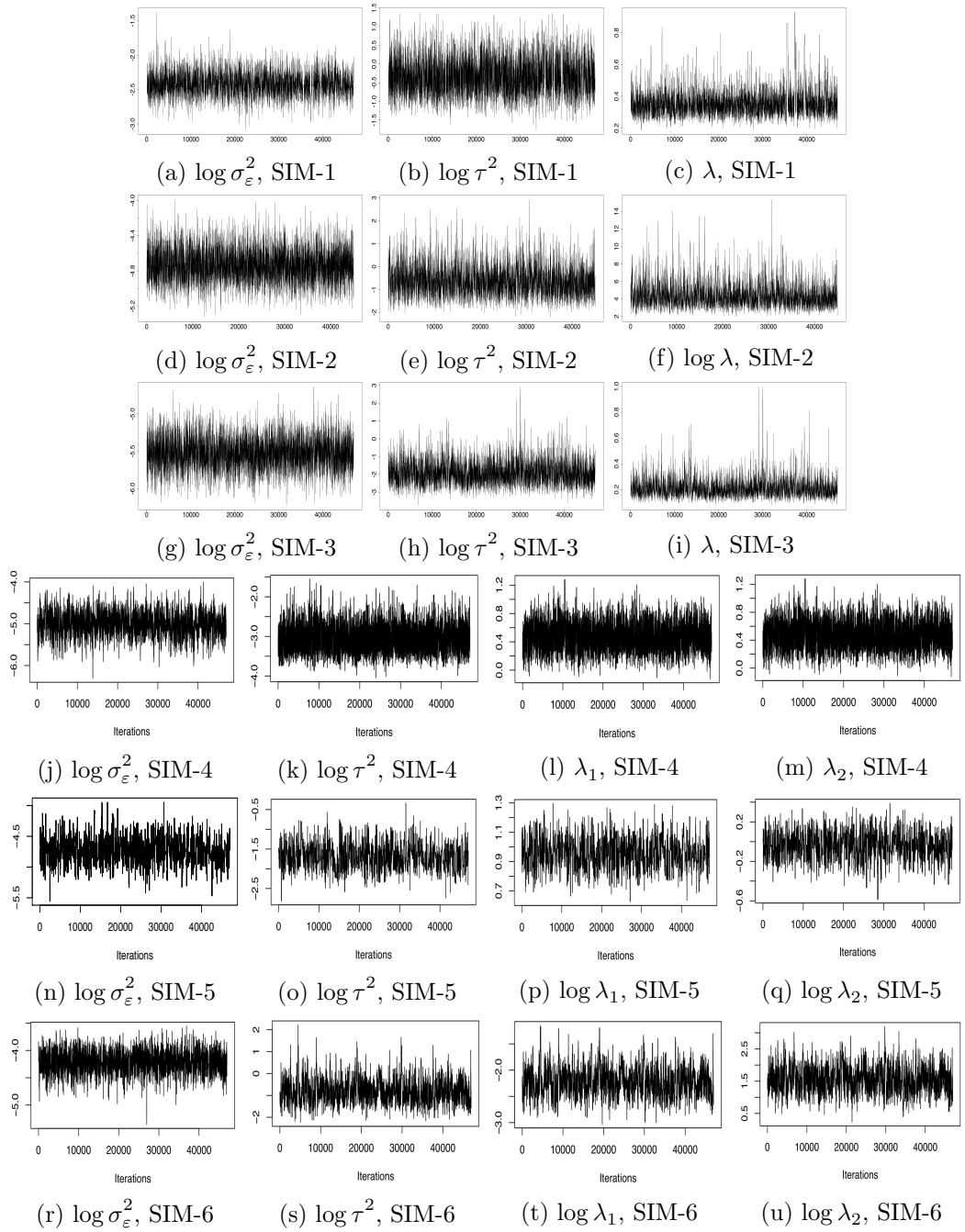
$$y(\boldsymbol{x}) = 2 + 0.01(x_1 - x_2)^2 + (1 - x_1) + 2(2 - x_2)^2 + 7\sin(0.5x_1)\sin(0.7x_1x_2) + \varepsilon,$$

where $\varepsilon \sim \mathrm{N}(0, 0.008)$.
SIM-6:

$$y(\boldsymbol{x}) = \sin\left(30(x_1 - 0.8)^4\right)\cos\left(2(x_2 - 0.6)^2\right) - 0.07(x_1^5 - x_2^3) - 0.3x_i^8 + x_1x_2^{12} + \varepsilon,$$

where $\varepsilon \sim \mathrm{N}(0, 0.008)$.

B.2 Convergence diagnosis



(a) $\log \sigma_\varepsilon^2$, SIM-1 (b) $\log \tau^2$, SIM-1 (c) λ, SIM-1

(d) $\log \sigma_\varepsilon^2$, SIM-2 (e) $\log \tau^2$, SIM-2 (f) $\log \lambda$, SIM-2

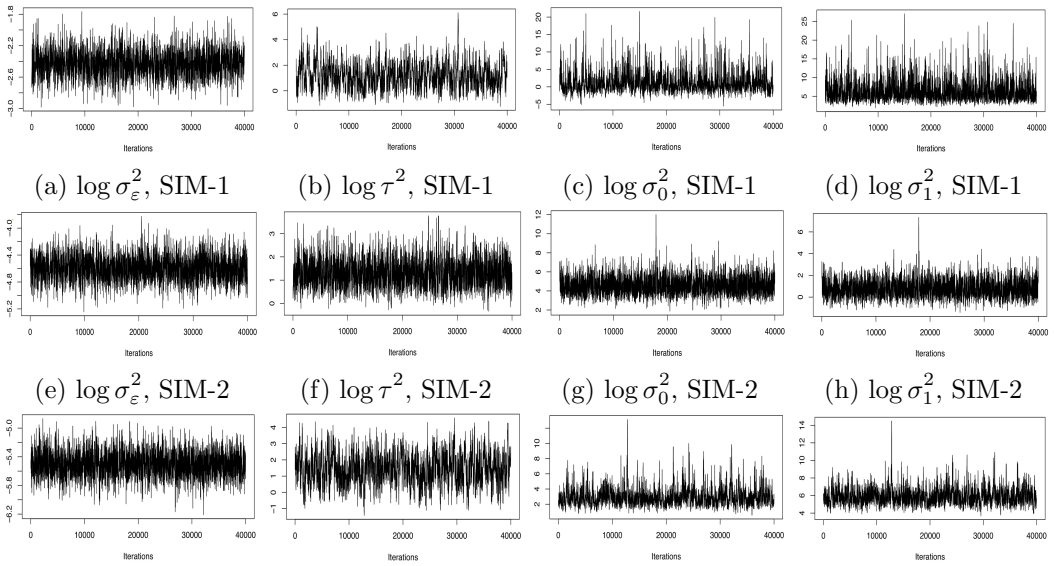
(g) $\log \sigma_\varepsilon^2$, SIM-3 (h) $\log \tau^2$, SIM-3 (i) λ, SIM-3

(j) $\log \sigma_\varepsilon^2$, SIM-4 (k) $\log \tau^2$, SIM-4 (l) λ_1, SIM-4 (m) λ_2, SIM-4

(n) $\log \sigma_\varepsilon^2$, SIM-5 (o) $\log \tau^2$, SIM-5 (p) $\log \lambda_1$, SIM-5 (q) $\log \lambda_2$, SIM-5

(r) $\log \sigma_\varepsilon^2$, SIM-6 (s) $\log \tau^2$, SIM-6 (t) $\log \lambda_1$, SIM-6 (u) $\log \lambda_2$, SIM-6

Figure B.1: Traceplots (after burnin and thinning) for 1-D and 2-D datasets with STAT model.

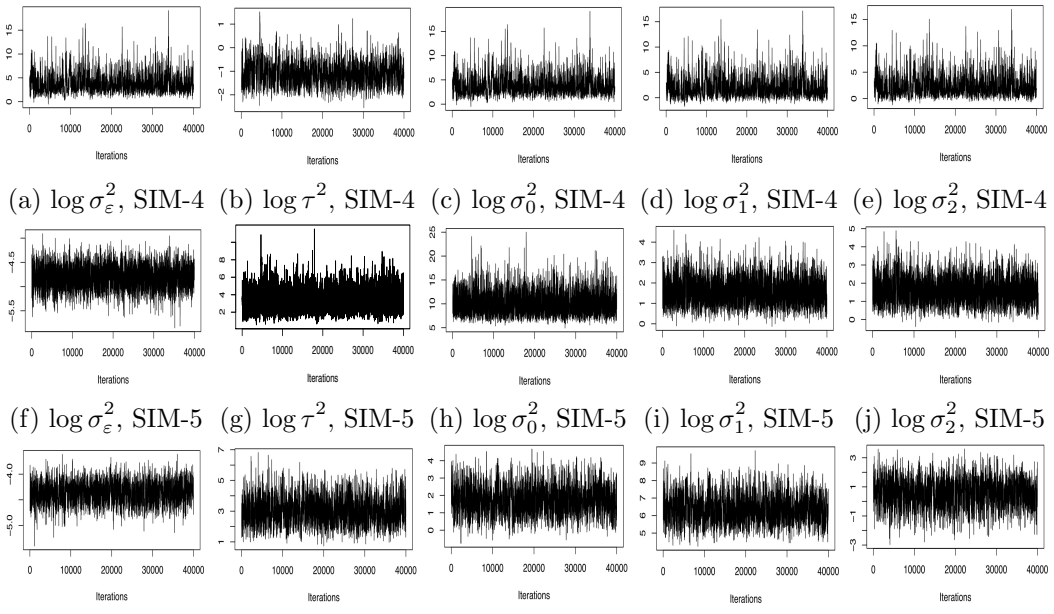


(a) $\log \sigma_\varepsilon^2$, SIM-1 (b) $\log \tau^2$, SIM-1 (c) $\log \sigma_0^2$, SIM-1 (d) $\log \sigma_1^2$, SIM-1

(e) $\log \sigma_\varepsilon^2$, SIM-2 (f) $\log \tau^2$, SIM-2 (g) $\log \sigma_0^2$, SIM-2 (h) $\log \sigma_1^2$, SIM-2

(i) $\log \sigma_\varepsilon^2$, SIM-3 (j) $\log \tau^2$, SIM-3 (k) $\log \sigma_0^2$, SIM-3 (l) $\log \sigma_1^2$, SIM-3

Figure B.2: Traceplots (after burnin and thinning) for 1-D datasets with NN model.

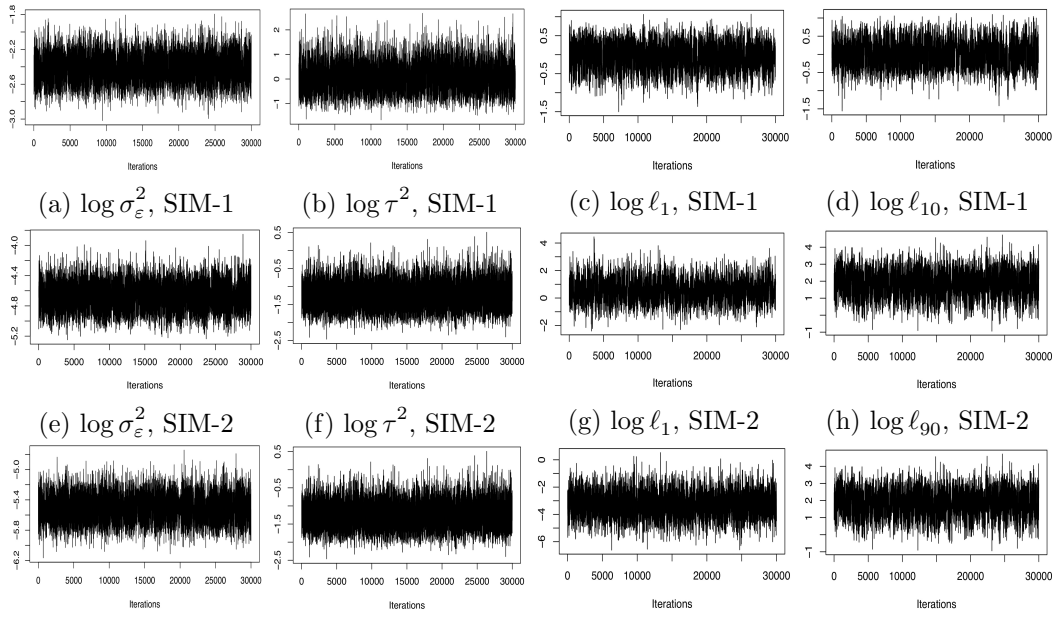


(a) $\log \sigma_\varepsilon^2$, SIM-4 (b) $\log \tau^2$, SIM-4 (c) $\log \sigma_0^2$, SIM-4 (d) $\log \sigma_1^2$, SIM-4 (e) $\log \sigma_2^2$, SIM-4

(f) $\log \sigma_\varepsilon^2$, SIM-5 (g) $\log \tau^2$, SIM-5 (h) $\log \sigma_0^2$, SIM-5 (i) $\log \sigma_1^2$, SIM-5 (j) $\log \sigma_2^2$, SIM-5

(k) $\log \sigma_\varepsilon^2$, SIM-6 (l) $\log \tau^2$, SIM-6 (m) $\log \sigma_0^2$, SIM-6 (n) $\log \sigma_1^2$, SIM-6 (o) $\log \sigma_2^2$, SIM-6

Figure B.3: Traceplots (after burnin and thinning) for 2-D datasets with NN model.

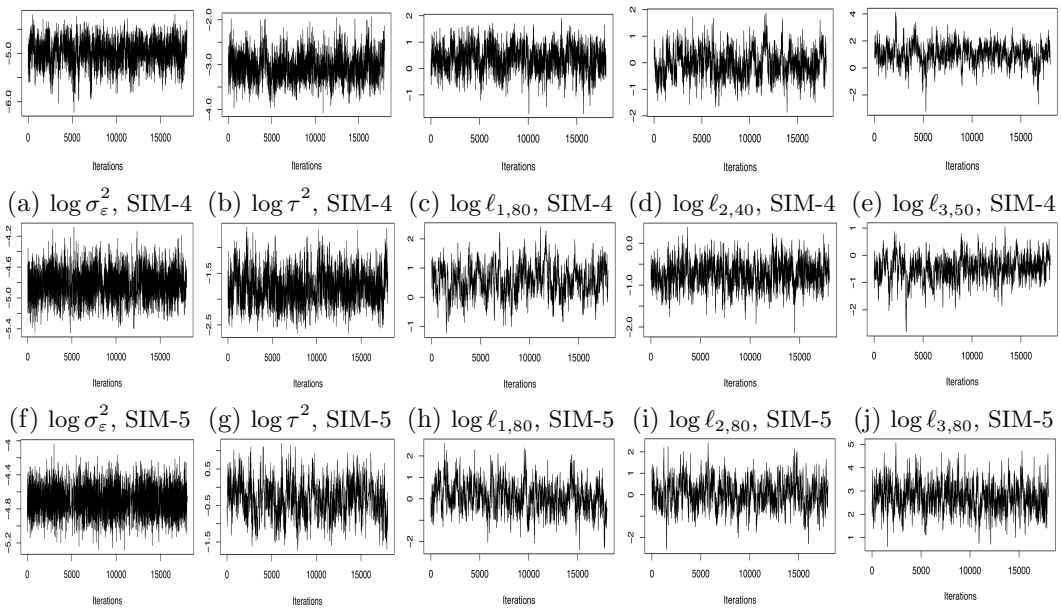


(a) $\log \sigma_\varepsilon^2$, SIM-1 (b) $\log \tau^2$, SIM-1 (c) $\log \ell_1$, SIM-1 (d) $\log \ell_{10}$, SIM-1

(e) $\log \sigma_\varepsilon^2$, SIM-2 (f) $\log \tau^2$, SIM-2 (g) $\log \ell_1$, SIM-2 (h) $\log \ell_{90}$, SIM-2

(i) $\log \sigma_\varepsilon^2$, SIM-3 (j) $\log \tau^2$, SIM-3 (k) $\log \ell_1$, SIM-3 (l) $\log \ell_{90}$, SIM-3

Figure B.4: Traceplots (after burnin) for some of the parameters in 1-D datasets with the 2-level model.



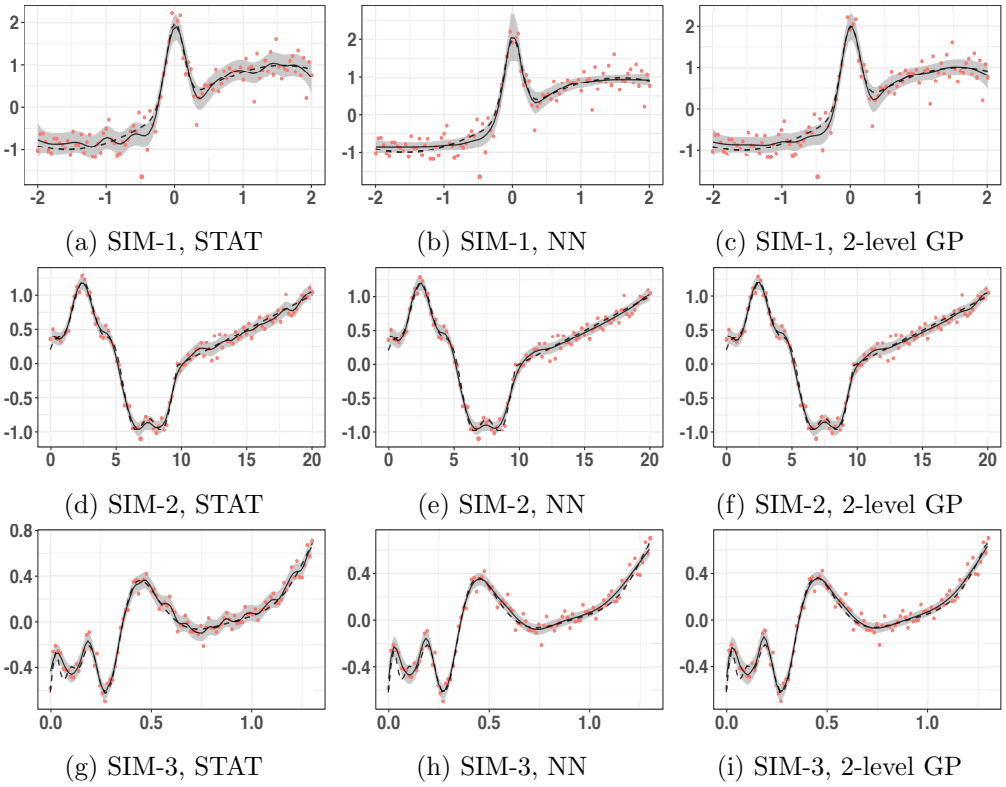
(a) $\log \sigma_\varepsilon^2$, SIM-4 (b) $\log \tau^2$, SIM-4 (c) $\log \ell_{1,80}$, SIM-4 (d) $\log \ell_{2,40}$, SIM-4 (e) $\log \ell_{3,50}$, SIM-4

(f) $\log \sigma_\varepsilon^2$, SIM-5 (g) $\log \tau^2$, SIM-5 (h) $\log \ell_{1,80}$, SIM-5 (i) $\log \ell_{2,80}$, SIM-5 (j) $\log \ell_{3,80}$, SIM-5

(k) $\log \sigma_\varepsilon^2$, SIM-6 (l) $\log \tau^2$, SIM-6 (m) $\log \ell_{1,90}$, SIM-6 (n) $\log \ell_{2,90}$, SIM-6 (o) $\log \ell_{3,90}$, SIM-6

Figure B.5: Traceplots (after burnin) for some of the parameters in 2-D datasets with the 2-level model.

B.3 Predictive performance

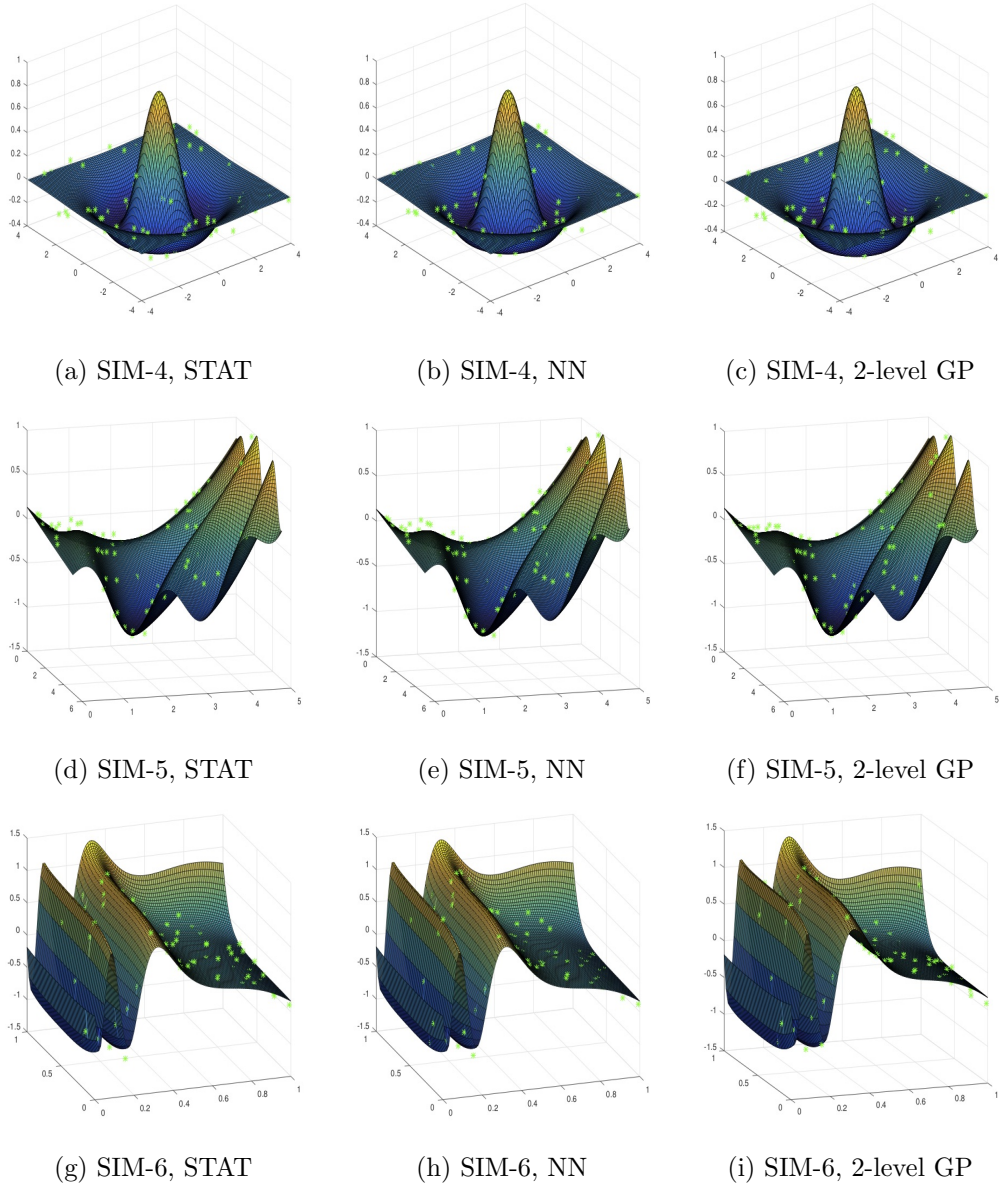


(a) SIM-1, STAT (b) SIM-1, NN (c) SIM-1, 2-level GP

(d) SIM-2, STAT (e) SIM-2, NN (f) SIM-2, 2-level GP

(g) SIM-3, STAT (h) SIM-3, NN (i) SIM-3, 2-level GP

Figure B.6: Predictions of the latent function for 1-D synthetic data. Dashed line shows the true function and black line depicts the posterior mean estimate. Grey area underlines 95% credible intervals. (a)-(c): Predictions for SIM-1. (d)-(f): Predictions for SIM-2. (g)-(i): Predictions for SIM-3.



(a) SIM-4, STAT (b) SIM-4, NN (c) SIM-4, 2-level GP

(d) SIM-5, STAT (e) SIM-5, NN (f) SIM-5, 2-level GP

(g) SIM-6, STAT (h) SIM-6, NN (i) SIM-6, 2-level GP

Figure B.7: Predictions of the latent function for 2-D synthetic data. Green crosses show the predicted value for the latent function at test locations. (a)-(c): Predictions for SIM-4. (d)-(f): Predictions for SIM-5. (g)-(i): Predictions for SIM-6.

APPENDIX C

SUPPLEMENTARY MATERIAL FOR CHAPTER 4

C.1 Experiments

We consider three simulated datasets with different characteristics. The first example is a function which has smooth parts and edges, and it is also piecewise constant,

$$z(x) = \begin{cases} \exp\left(4 - \frac{25}{x(5-x)}\right) & x \in (0,5) \\ 1 & x \in [7,8] \\ -1 & x \in (8,9] \\ 0 & \text{otherwise} \end{cases}.$$

The second corresponds to a damped sine wave function,

$$z(x) = \exp(-x)\cos(2\pi x).$$

The third employs the *Bumps* function in Donoho and Johnstone (1995) with the data scaled to have zero mean and unit variance. Following Vannucci and Corradi (1999), we generate $N = 512$ points in the interval [0,1] and use a signal-to-noise ratio equal to 5, such that the noise variance $\sigma_\varepsilon^2 = 0.04$.

C.1.1 Experiment 1

		MWG			w-ELL-SS			m-ELL-SS		
		$M = 85$	$M = 169$	$M = 253$	$M = 85$	$M = 169$	$M = 253$	$M = 85$	$M = 169$	$M = 253$
AR(1)	σ_ε^2	0.014	0.015	0.015	0.014	0.014	0.014	0.014	0.014	0.014
	ℓ_j	2.416	2.653	2.785	2.350	1.912	2.015	2.118	2.163	1.968
	z_j	0.687	0.686	0.685	0.690	0.693	0.693	0.692	0.692	0.692
	λ	0.435	0.405	0.385	0.312	0.408	0.379	0.381	0.358	0.338
SE	σ_ε^2	0.031	0.043	0.055	0.013	0.013	0.015	0.013	0.013	0.013
	ℓ_j	0.678	1.183	1.165	1.888	2.147	1.709	2.119	2.142	2.145
	z_j	0.690	0.698	0.674	0.692	0.692	0.691	0.692	0.693	0.693
	λ	0.543	0.545	0.539	0.188	0.191	0.476	0.186	0.181	0.174

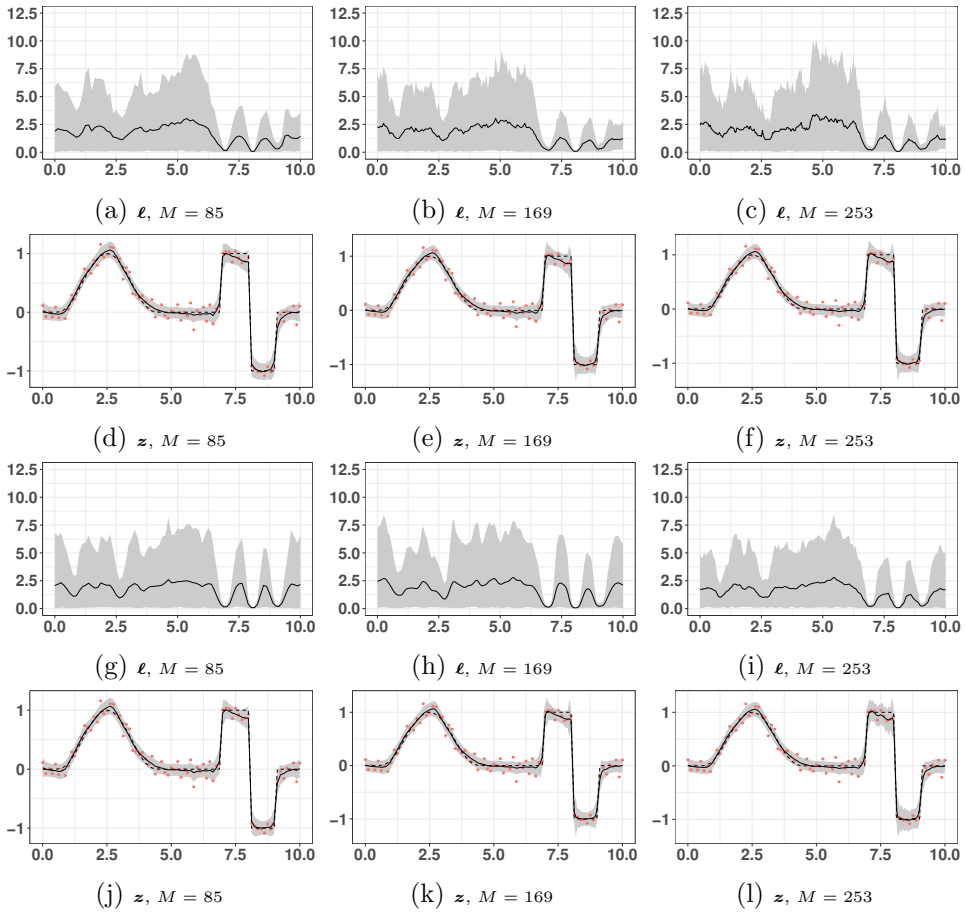
Table C.1: Experiment 1: Posterior mean estimates with both hyperpriors under various discretisation schemes ($M = 85, 169, 253$) and three different algorithms.

		AR(1)			SE		
		Burned	Non-burned	Total time	Burned	Non-burned	Total time
MWG	$M = 85$	**0.01**	16.78	16.80	28.02	NA	28.02
	$M = 169$	0.04	**40.66**	**40.69**	103.55	NA	103.55
	$M = 253$	0.10	**76.84**	**76.94**	265.16	NA	265.16
w-ELL-SS	$M = 85$	0.04	**14.55**	**14.58**	0.18	24.84	25.02
	$M = 169$	0.30	51.90	52.20	0.82	103.05	103.86
	$M = 253$	0.70	127.67	128.37	3.22	249.15	252.37
m-ELL-SS	$M = 85$	**0.01**	18.50	18.52	**0.03**	**22.17**	**22.20**
	$M = 169$	**0.03**	46.54	46.57	**0.18**	**59.37**	**59.60**
	$M = 253$	**0.06**	104.20	104.26	**0.37**	**132.97**	**133.35**

Table C.2: Experiment 1: CPU time (minutes) for 200,000 iterations. NA denotes that MWG for the SE hyperprior did not converge. Best values in boldface.

		MWG			w-ELL-SS			m-ELL-SS		
		$M = 85$	$M = 169$	$M = 253$	$M = 85$	$M = 169$	$M = 253$	$M = 85$	$M = 169$	$M = 253$
AR(1)	σ_ε^2	10452.5	7038.0	5070.5	5541.0	5313.1	4967.9	**12234.4**	**11999.4**	**12124.1**
	ℓ_{15}	**5424.4**	2150.5	1317.3	181.1	192.0	201.6	3146.7	**3391.2**	**3282.9**
	ℓ_{66}	**22539.7**	**11131.5**	**6901.8**	773.2	467.1	268.0	9337.0	3736.3	3557.9
	z_{15}	25449.8	11648.3	7878.2	4635.0	5981.3	5264.1	**30601.6**	**35096.7**	**47895.6**
	z_{66}	**42146.1**	27135.4	21528.7	8343.2	7485.8	8127.4	**26530.5**	**27856.4**	**26881.2**
	λ	1507.9	636.6	460.8	331.2	272.8	300.9	**2068.6**	**2119.5**	**2243.5**
SE	σ_ε^2	313.4	505.5	1986.6	6117.6	8008.2	2214.8	**18983.7**	**15087.8**	**16750.4**
	ℓ_{15}	2.1	7.5	6.7	214.0	195.7	289.1	**3401.0**	**3498.8**	**3381.2**
	ℓ_{66}	2.1	2.1	1.4	961.5	717.8	309.1	**8434.1**	**7023.2**	**7391.8**
	z_{15}	**91330.7**	22391.1	117111.0	4992.2	5113.6	5989.6	28060.0	**30737.0**	**28382.6**
	z_{66}	48.4	83.1	8678.6	11139.8	12676.2	2561.6	**31456.3**	**33268.2**	**41623.7**
	λ	16.6	77.4	82.3	57.5	29.5	3.6	**367.8**	**246.9**	**293.3**

Table C.3: Results Experiment 1: ESS after burn-in period for both hyperpriors under various discretisation schemes ($M = 85, 169, 253$) and employing three different sampling algorithms. Highest values in boldface.



(a) $\boldsymbol{\ell}$, $M = 85$ (b) $\boldsymbol{\ell}$, $M = 169$ (c) $\boldsymbol{\ell}$, $M = 253$

(d) $\boldsymbol{z}$, $M = 85$ (e) $\boldsymbol{z}$, $M = 169$ (f) $\boldsymbol{z}$, $M = 253$

(g) $\boldsymbol{\ell}$, $M = 85$ (h) $\boldsymbol{\ell}$, $M = 169$ (i) $\boldsymbol{\ell}$, $M = 253$

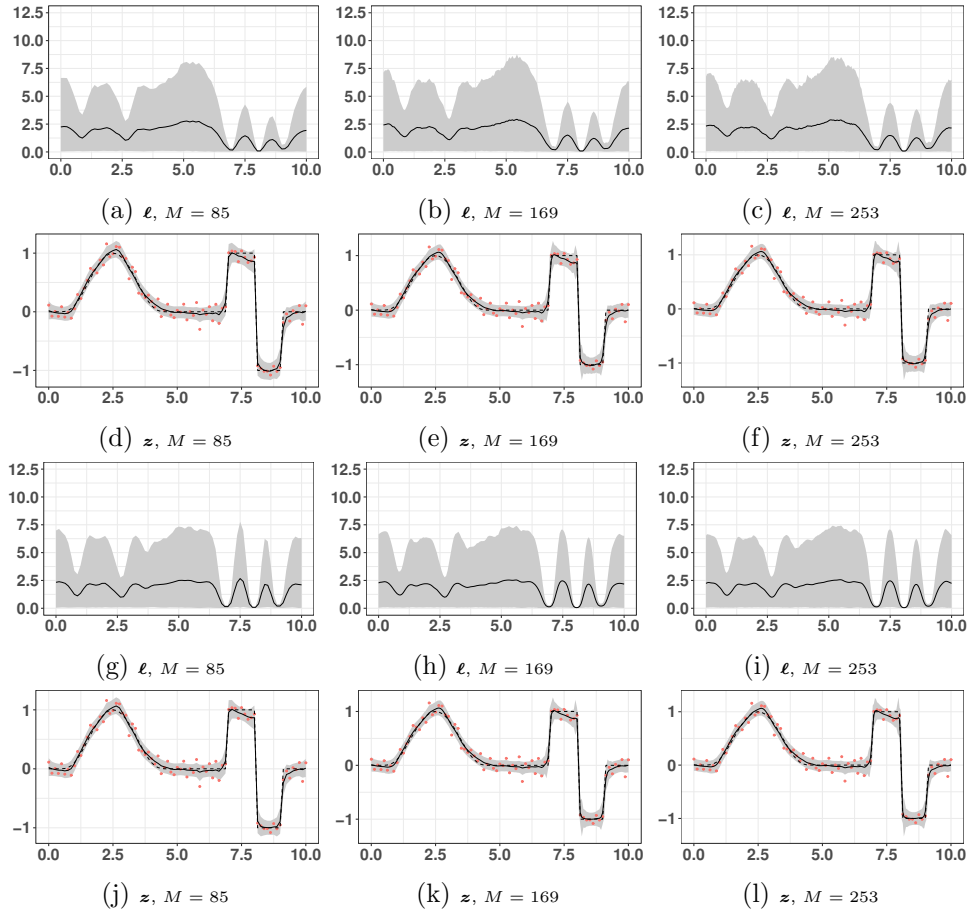
(j) $\boldsymbol{z}$, $M = 85$ (k) $\boldsymbol{z}$, $M = 169$ (l) $\boldsymbol{z}$, $M = 253$

Figure C.1: Experiment 1 with w-ELL-SS algorithm. (a)-(c): Estimated $\boldsymbol{\ell}$ process with 95% credible intervals for AR(1) hyperprior on different grids. (d)-(f): Estimated $\boldsymbol{z}$ process with 95% credible intervals for AR(1) hyperprior on different grids with observed data in red. (g)-(i): Estimated $\boldsymbol{\ell}$ process with 95% credible intervals for SE hyperprior on different grids. (j)-(l): Estimated $\boldsymbol{z}$ process with 95% credible intervals for SE hyperprior on different grids with observed data in red.

C.1.2 Experiment 2

		AR(1)			SE		
		Burned	Non-burned	Total time	Burned	Non-burned	Total time
MWG	$M = 430$	0.60	**155.82**	**156.43**	572.36	NA	572.36
w-ELL-SS	$M = 430$	1.42	306.60	308.02	3.60	500.49	504.09
m-ELL-SS	$M = 430$	**0.25**	308.67	308.92	**1.17**	**330.04**	**331.22**

Table C.4: Experiment 2: CPU time (minutes) for 100,000 iterations. NA denotes that MWG for SE hyperprior did not converge. Best values in boldface.



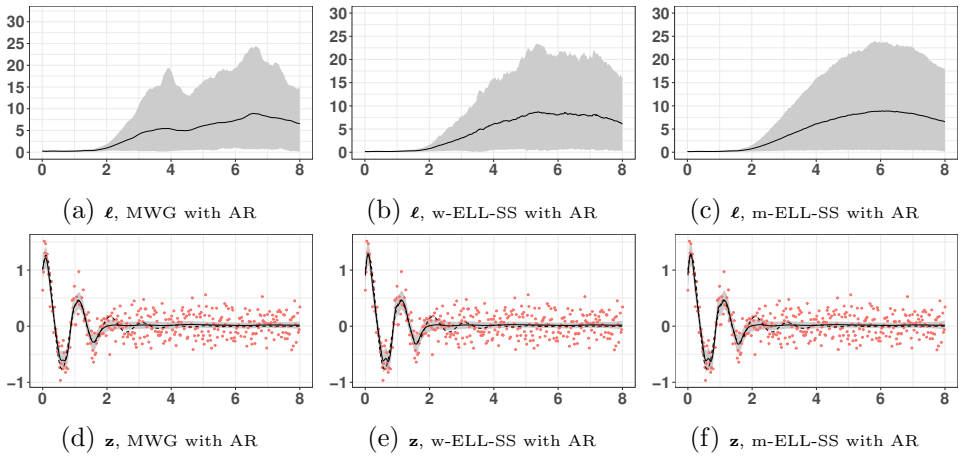
(a) $\boldsymbol{\ell}$, $M = 85$ (b) $\boldsymbol{\ell}$, $M = 169$ (c) $\boldsymbol{\ell}$, $M = 253$

(d) $\boldsymbol{z}$, $M = 85$ (e) $\boldsymbol{z}$, $M = 169$ (f) $\boldsymbol{z}$, $M = 253$

(g) $\boldsymbol{\ell}$, $M = 85$ (h) $\boldsymbol{\ell}$, $M = 169$ (i) $\boldsymbol{\ell}$, $M = 253$

(j) $\boldsymbol{z}$, $M = 85$ (k) $\boldsymbol{z}$, $M = 169$ (l) $\boldsymbol{z}$, $M = 253$

Figure C.2: Experiment 1 with m-ELL-SS algorithm. (a)-(c): Estimated $\boldsymbol{\ell}$ process with 95% credible intervals for AR(1) hyperprior on different grids. (d)-(f): Estimated $\boldsymbol{z}$ process with 95% credible intervals for AR(1) hyperprior on different grids with observed data in red.. (g)-(i): Estimated $\boldsymbol{\ell}$ process with 95% credible intervals for SE hyperprior on different grids. (j)-(l): Estimated $\boldsymbol{z}$ process with 95% credible intervals for SE hyperprior on different grids with observed data in red.



(a) $\boldsymbol{\ell}$, MWG with AR (b) $\boldsymbol{\ell}$, w-ELL-SS with AR (c) $\boldsymbol{\ell}$, m-ELL-SS with AR

(d) $\mathbf{z}$, MWG with AR (e) $\mathbf{z}$, w-ELL-SS with AR (f) $\mathbf{z}$, m-ELL-SS with AR

Figure C.3: Experiment 2 for AR hyperprior and different samplers. Top row: estimated $\boldsymbol{\ell}$ process with 95% credible interval for AR(1) hyperprior with (a) MWG, (b) w-ELL-SS and (c) m-ELL-SS. Second row: estimated $\mathbf{z}$ process with 95% credible interval for AR(1) hyperprior with (d) MWG, (e) w-ELL-SS and (f) m-ELL-SS.

		MWG	w-ELL-SS	m-ELL-SS
AR(1)	σ^2_ε	0.045	0.044	0.044
	ℓ_{100}	1.694	1.379	1.287
	ℓ_{200}	5.051	6.922	7.131
	z_{100}	0.021	0.025	0.027
	z_{200}	0.031	0.027	0.027
	λ	2.598	2.771	2.710
SE	σ^2_ε	0.072	0.044	0.044
	ℓ_{100}	0.594	0.965	.951
	ℓ_{200}	0.677	8.967	9.187
	z_{100}	0.032	0.029	0.029
	z_{200}	0.060	0.025	0.024
	λ	0.450	1.877	1.970

Table C.5: Experiment 2: Posterior mean estimates obtained with both hyperpriors and employing three different sampling algorithms. Estimates are consistent across sampling algorithms, except for SE with MWG because the sampler did not reach convergence.

C.1.3 Experiment 3

C.1.3.1 Prior elicitation

As opposed to Experiment 1 and 2, where vague priors for covariance parameters sufficed, here we employ informative prior distributions for $\log(\lambda)$ and $\boldsymbol{u}$. Knowledge about the parameters comes from the fact that the length-scales, for both stationary

		MWG	w-ELL-SS	m-ELL-SS
AR(1)	σ_ε^2	14505.3	17446.5	**20673.4**
	ℓ_{100}	116.3	282.6	**2485.3**
	ℓ_{200}	56.3	385.5	**2421.7**
	z_{100}	7002.5	13637.9	**37023.4**
	z_{200}	3424.5	8179.6	**27585.5**
	λ	92.6	145.7	**1312.8**
SE	σ_ε^2	444.5	18804.2	**21169.3**
	ℓ_{100}	5.0	1145.9	**5996.4**
	ℓ_{200}	7.4	919.4	**3563.6**
	z_{100}	**100000.0**	37550.5	76574.
	z_{200}	**98891.7**	14476.0	49195.2
	λ	44.8	91.0	**668.4**

Table C.6: Experiment 2: ESS after burnin period for both hyperprior and employing three different sampling algorithms. Highest values in boldface. m-ELL-SS results in the highest efficiency scores.

		MWG	w-ELL-SS	m-ELL-SS
AR(1)	σ_ε^2	0.041	0.040	0.040
	ℓ_{100}	1.821	3.780	1.520
	ℓ_{200}	0.519	0.375	0.510
	z_{100}	-0.519	-0.538	-0.535
	z_{200}	2.097	2.110	2.086
	λ	0.033	0.029	0.033
SE	σ_ε^2	0.504	0.039	0.039
	ℓ_{100}	1.414	0.126	0.666
	ℓ_{200}	1.523	0.310	0.381
	z_{100}	0.178	-0.499	-0.523
	z_{200}	1.303	2.046	2.053
	λ	1.058	0.106	0.024

Table C.7: Experiment 3: Posterior mean estimates obtained with both hyperpriors and employing three different sampling algorithms.

		MWG	w-ELL-SS	m-ELL-SS
AR(1)	σ_ε^2	**6975.6**	3398.3	4638.5
	ℓ_{100}	**489.5**	8.2	155.1
	ℓ_{200}	**1978.3**	63.3	201.8
	z_{100}	**6875.2**	3354.2	5220.6
	z_{200}	**4515.2**	817.0	910.6
	λ	**193.4**	18.4	106.1
SE	σ_ε^2	2650.1	5072.4	**12442.0**
	ℓ_{100}	2.4	70.	**153.7**
	ℓ_{200}	2.5	310.3	**1339.5**
	z_{100}	3522.7	49136.2	**6397.9**
	z_{200}	2101.0	36809.1	**4399.9**
	λ	**93.4**	2.5	27.2

Table C.8: Results for Experiment 3: ESS after burnin period for both hyperprior and employing three different sampling algorithms. Highest values in boldface.

		AR(1)			SE		
		Burned	Non-burned	Total time	Burned	Non-burned	Total time
MWG	$M = 572$	32.48	**297.78**	**330.27**	1289.166	NA	1289.166
w-ELL-SS	$M = 572$	106.43	592.95	699.38	6.02	1246.85	1252.87
m-ELL-SS	$M = 572$	**20.70**	814.10	834.79	**85.17**	**810.19**	**895.36**

Table C.9: Experiment 3: CPU time (minutes) for 100,000 iterations. NA denotes that MWG for SE hyperprior did not converge. Best values in boldface.

		AR(1)			SE		
		Burned	Non-burned	Total time	Burned	Non-burned	Total time
MWG	$M = 572$	27.86	**249.14**	**277.00**	956.78	NA	956.78
w-ELL-SS	$M = 572$	45.91	258.61	304.52	402.77	NA	**402.77**
m-ELL-SS	$M = 572$	**9.39**	375.90	385.29	**42.98**	**397.12**	440.10

Table C.10: Computational time for Experiment 3 in a high performance computer. Algorithms were run for 100,000 iterations. m-ELL-SS and w-ELL-SS speed up by a factor of approximately 2.1, while MWG by 1.1.

and non-stationary processes, are only identifiable between the minimum and maximum covariate distance. In this experiment, the maximum distance is 1 and the minimum is .0019; thus, the $\mathrm{N}(0, 1)$ prior for each u_j is inappropriate. Instead, we solve the system of equations of Section 3.4 (Chapter 3) to fix the hyperparameters. Indeed, arbitrarily fixing the hyperparameters can greatly affect the inferences. See for instance the estimated length-scale process with MWG and AR hyperprior in Figure C.4, where we set the prior of $\boldsymbol{u}$ to be a zero-centred GP with unit variance.

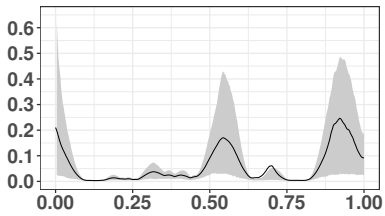
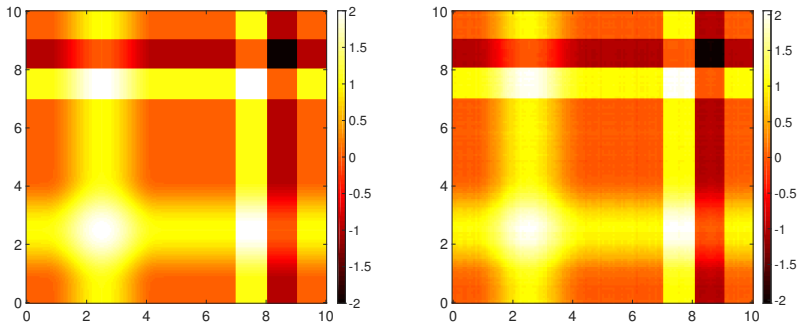


Figure C.4: Posterior mean of lengh-scale for Experiment 3 with MGW and AR hyperprior with $\mu_u = 0$ and $\tau_u^2 = 1$.

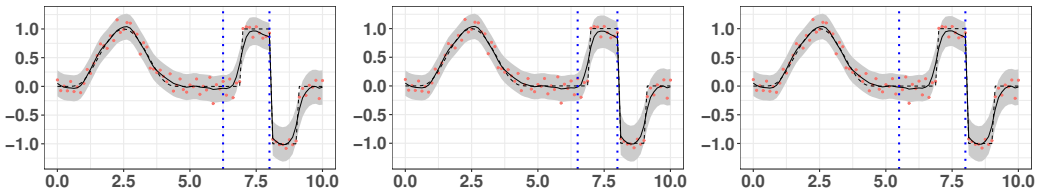


(a) True (b) Posterior mean

Figure C.5: True vs. posterior mean for two-dimensional simulated dataset.

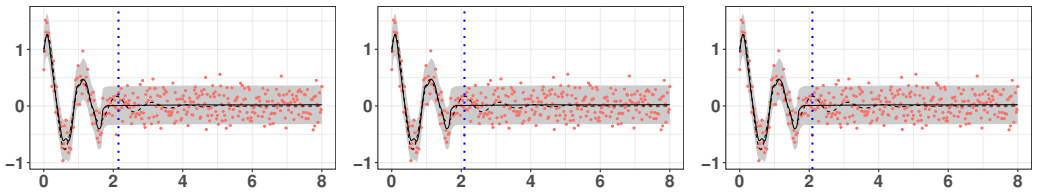
C.1.4 Two-dimensional synthetic data

C.2 Comparative Evaluation



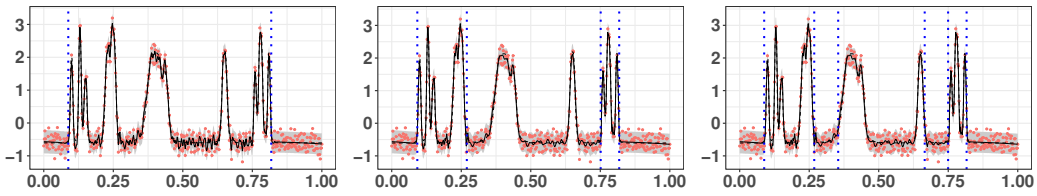
(a) MAE=0.056 (b) MAE=.057 (c) MAE=.057

Figure C.6: TGP model results for Experiment 1 with different chain lengths. (a): 100, 000 iterations with 20, 000 burn-in. (b): 200, 000 iterations with 50, 000 burn-in. (c): 500, 000 iterations with 100, 000 burn-in.



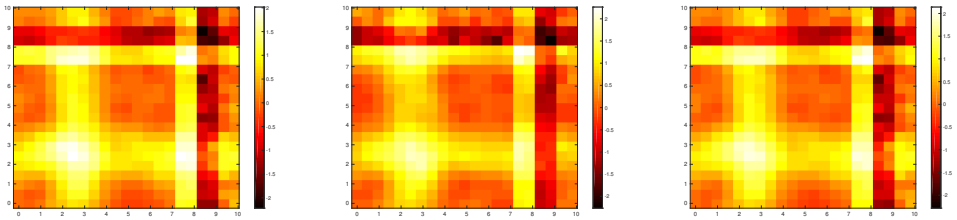
(a) MAE=0.043 (b) MAE=0.043 (c) MAE=.043

Figure C.7: TGP model results for Experiment 2 with different chain lengths. (a): 100, 000 iterations with 20, 000 burn-in. (b): 200, 000 iterations with 50, 000 burn-in. (c): 500, 000 iterations with 100, 000 burn-in.



(a) MAE=0.079 (b) MAE=.067 (c) MAE=.065

Figure C.8: TGP model results for Experiment 3 with different chain lengths. (a): 100,000 iterations with 20,000 burn-in. (b): 200,000 iterations with 50,000 burn-in. (c): 500,000 iterations with 100,000 burn-in and thinning of 5. Increasing the number of iterations has a positive effect on the number of partitions found. However, without knowing the ground truth, it is hard to know beforehand if the algorithm has been run for long enough to find the appropriate number of partitions.



(a) MAE=0.122 (b) MAE=.131 (c) MAE=.123

Figure C.9: TGP model results for Experiment 4 (subset) with different chain lengths. (a): 100,000 iterations with 20,000 burn-in. (b): 200,000 iterations with 50,000 burn-in. (c): 500,000 iterations with 100,000 burn-in and thinning of 5.

APPENDIX D

SUPPLEMENTARY MATERIAL FOR CHAPTER 5

D.1 Derivation of the marginal variational posterior

Consider the optimal variational posterior for the ARD case:

$$q(\widetilde{\boldsymbol{z}}, \widetilde{U}, \boldsymbol{\theta}, \boldsymbol{\varphi}) \propto \left(\frac{\sigma_\varepsilon^{-2}}{2\pi}\right)^{\frac{N}{2}} \pi(\widetilde{\boldsymbol{z}}|\widetilde{U}, \tau_z^2, \psi)\pi(\boldsymbol{\theta}) \left(\prod_{d=1}^{D} \pi(\widetilde{\boldsymbol{u}}_{:d} \mid \boldsymbol{\varphi}_d)\pi(\boldsymbol{\varphi}_d)\right) \times$$
$$\exp\left(-\frac{1}{2\sigma_\varepsilon^2}\sum_{n=1}^{N}(y_n - \boldsymbol{\beta}_n(C^{\text{NS}}_{\widetilde{\boldsymbol{z}},\widetilde{\boldsymbol{z}}})^{-1}\widetilde{\boldsymbol{z}})^2 + \tau_z^2 - \alpha_n + \widetilde{\boldsymbol{z}}^T(C^{\text{NS}}_{\widetilde{\boldsymbol{z}},\widetilde{\boldsymbol{z}}})^{-1}(P_n - \boldsymbol{\beta}_n^T\boldsymbol{\beta}_n)(C^{\text{NS}}_{\widetilde{\boldsymbol{z}},\widetilde{\boldsymbol{z}}})^{-1}\widetilde{\boldsymbol{z}}\right). \tag{D.1}$$

Our goal is to marginalse $\widetilde{\boldsymbol{z}}$ to obtain $q(\widetilde{U}, \boldsymbol{\theta}, \boldsymbol{\varphi})$ and consequently the conditional variational posterior $q(\widetilde{\boldsymbol{z}} \mid \boldsymbol{\theta}, \boldsymbol{\varphi})$.

First, by expanding the terms inside the exponent

$$q(\widetilde{\boldsymbol{z}}, \widetilde{U}, \boldsymbol{\theta}, \boldsymbol{\varphi}) \propto \left(\frac{\sigma_\varepsilon^{-2}}{2\pi}\right)^{\frac{N}{2}} \pi(\widetilde{\boldsymbol{z}}|\widetilde{U}, \tau_z^2, \psi)\pi(\boldsymbol{\theta}) \left(\prod_{d=1}^{D} \pi(\widetilde{\boldsymbol{u}}_{:d} \mid \boldsymbol{\varphi}_d)\pi(\boldsymbol{\varphi}_d)\right) \times$$
$$\exp\left(-\frac{1}{2\sigma_\varepsilon^2}\sum_{n=1}^{N} y_n^2 - 2y_n\boldsymbol{\beta}_n(C^{\text{NS}}_{\widetilde{\boldsymbol{z}},\widetilde{\boldsymbol{z}}})^{-1}\widetilde{\boldsymbol{z}} + \cancel{\boldsymbol{\beta}_n(C^{\text{NS}}_{\widetilde{\boldsymbol{z}},\widetilde{\boldsymbol{z}}})^{-1}\widetilde{\boldsymbol{z}}\left(\boldsymbol{\beta}_n(C^{\text{NS}}_{\widetilde{\boldsymbol{z}},\widetilde{\boldsymbol{z}}})^{-1}\widetilde{\boldsymbol{z}}\right)^T} + \tau_z^2 - \alpha_n + \right.$$
$$\left. \widetilde{\boldsymbol{z}}^T(C^{\text{NS}}_{\widetilde{\boldsymbol{z}},\widetilde{\boldsymbol{z}}})^{-1}P_n(C^{\text{NS}}_{\widetilde{\boldsymbol{z}},\widetilde{\boldsymbol{z}}})^{-1}\widetilde{\boldsymbol{z}} - \cancel{\widetilde{\boldsymbol{z}}^T(C^{\text{NS}}_{\widetilde{\boldsymbol{z}},\widetilde{\boldsymbol{z}}})^{-1}(\boldsymbol{\beta}_n^T\boldsymbol{\beta}_n)(C^{\text{NS}}_{\widetilde{\boldsymbol{z}},\widetilde{\boldsymbol{z}}})^{-1}\widetilde{\boldsymbol{z}}}\right),$$

and collecting the terms that do not depend on $\boldsymbol{z}$ in

$$\Xi := \left(\frac{\sigma_\varepsilon^{-2}}{2\pi}\right)^{\frac{N}{2}} \pi(\boldsymbol{\theta}) \left(\prod_{d=1}^{D} \pi(\widetilde{\boldsymbol{u}}_{:d} \mid \boldsymbol{\varphi}_d)\pi(\boldsymbol{\varphi}_d)\right) \exp\left(-\frac{1}{2\sigma_\varepsilon^2}\sum_{n=1}^{N} y_n^2 + \tau_z^2 - \alpha_n\right),$$

we re-write Eq. (D.1) as

$$\begin{aligned}
&q(\widetilde{\boldsymbol{z}}, \widetilde{U}, \boldsymbol{\theta}, \boldsymbol{\varphi}) \\
&\propto \Xi \exp\left(-\frac{1}{2\sigma_\varepsilon^2}\sum_{n=1}^{N} -2y_n\boldsymbol{\beta}_n(C^{\text{NS}}_{\widetilde{\boldsymbol{z}},\widetilde{\boldsymbol{z}}})^{-1}\widetilde{\boldsymbol{z}} + \widetilde{\boldsymbol{z}}^T(C^{\text{NS}}_{\widetilde{\boldsymbol{z}},\widetilde{\boldsymbol{z}}})^{-1}P_n(C^{\text{NS}}_{\widetilde{\boldsymbol{z}},\widetilde{\boldsymbol{z}}})^{-1}\widetilde{\boldsymbol{z}}\right) \pi(\widetilde{\boldsymbol{z}}|\widetilde{U}, \tau_z^2, \psi) \\
&\propto \Xi \exp\left(-\frac{1}{2\sigma_\varepsilon^2}\sum_{n=1}^{N} -2y_n\boldsymbol{\beta}_n(C^{\text{NS}}_{\widetilde{\boldsymbol{z}},\widetilde{\boldsymbol{z}}})^{-1}\widetilde{\boldsymbol{z}} + \widetilde{\boldsymbol{z}}^T(C^{\text{NS}}_{\widetilde{\boldsymbol{z}},\widetilde{\boldsymbol{z}}})^{-1}P_n(C^{\text{NS}}_{\widetilde{\boldsymbol{z}},\widetilde{\boldsymbol{z}}})^{-1}\widetilde{\boldsymbol{z}}\right) \text{N}(\widetilde{\boldsymbol{z}}\mid, 0, C^{\text{NS}}_{\widetilde{\boldsymbol{z}},\widetilde{\boldsymbol{z}}}) \\
&\propto \Xi \exp\left(\sigma_\varepsilon^{-2}\boldsymbol{y}^T B(C^{\text{NS}}_{\widetilde{\boldsymbol{z}},\widetilde{\boldsymbol{z}}})^{-1}\widetilde{\boldsymbol{z}} - \frac{1}{2}\widetilde{\boldsymbol{z}}^T(C^{\text{NS}}_{\widetilde{\boldsymbol{z}},\widetilde{\boldsymbol{z}}})^{-1}\sigma_\varepsilon^{-2}\boldsymbol{P}(C^{\text{NS}}_{\widetilde{\boldsymbol{z}},\widetilde{\boldsymbol{z}}})^{-1}\widetilde{\boldsymbol{z}}\right) \text{N}(\widetilde{\boldsymbol{z}} \mid 0, C^{\text{NS}}_{\widetilde{\boldsymbol{z}},\widetilde{\boldsymbol{z}}}),
\end{aligned}$$

where $\boldsymbol{P} = \sum_{n=1}^{N} P_n$ and B an $(N \times M)$ matrix with rows $\boldsymbol{\beta}_n$.

The marginal variational posterior $q(\widetilde{U}, \boldsymbol{\theta}, \boldsymbol{\varphi}) = \int q(\widetilde{\boldsymbol{z}}, \widetilde{U}, \boldsymbol{\theta}, \boldsymbol{\varphi})\, d\widetilde{\boldsymbol{z}}$ is therefore:

$$\begin{aligned}
&q(\widetilde{U}, \boldsymbol{\theta}, \boldsymbol{\varphi}) \\
&\propto \int \Xi \exp\left(\sigma_\varepsilon^{-2}\boldsymbol{y}^T B(C^{\text{NS}}_{\widetilde{\boldsymbol{z}},\widetilde{\boldsymbol{z}}})^{-1}\widetilde{\boldsymbol{z}} - \frac{1}{2}\widetilde{\boldsymbol{z}}^T(C^{\text{NS}}_{\widetilde{\boldsymbol{z}},\widetilde{\boldsymbol{z}}})^{-1}\sigma_\varepsilon^{-2}\boldsymbol{P}(C^{\text{NS}}_{\widetilde{\boldsymbol{z}},\widetilde{\boldsymbol{z}}})^{-1}\widetilde{\boldsymbol{z}}\right) \text{N}(\widetilde{\boldsymbol{z}} \mid 0, C^{\text{NS}}_{\widetilde{\boldsymbol{z}},\widetilde{\boldsymbol{z}}})\, d\widetilde{\boldsymbol{z}} \\
&\propto \Xi|C^{\text{NS}}_{\widetilde{\boldsymbol{z}},\widetilde{\boldsymbol{z}}}|^{-\frac{1}{2}} \int \exp\left(\sigma_\varepsilon^{-2}\boldsymbol{y}^T B(C^{\text{NS}}_{\widetilde{\boldsymbol{z}},\widetilde{\boldsymbol{z}}})^{-1}\widetilde{\boldsymbol{z}} - \frac{1}{2}\widetilde{\boldsymbol{z}}^T(C^{\text{NS}}_{\widetilde{\boldsymbol{z}},\widetilde{\boldsymbol{z}}})^{-1}\sigma_\varepsilon^{-2}\boldsymbol{P}(C^{\text{NS}}_{\widetilde{\boldsymbol{z}},\widetilde{\boldsymbol{z}}})^{-1}\widetilde{\boldsymbol{z}}\right. \\
&\qquad\qquad \left. -\frac{1}{2}\widetilde{\boldsymbol{z}}^T(C^{\text{NS}}_{\widetilde{\boldsymbol{z}},\widetilde{\boldsymbol{z}}})^{-1}\widetilde{\boldsymbol{z}}\right)\, d\widetilde{\boldsymbol{z}} \\
&\propto \Xi|C^{\text{NS}}_{\widetilde{\boldsymbol{z}},\widetilde{\boldsymbol{z}}}|^{-\frac{1}{2}} \int \exp\left(\sigma_\varepsilon^{-2}\boldsymbol{y}^T B(C^{\text{NS}}_{\widetilde{\boldsymbol{z}},\widetilde{\boldsymbol{z}}})^{-1}\widetilde{\boldsymbol{z}} - \frac{1}{2}\widetilde{\boldsymbol{z}}^T\left[(C^{\text{NS}}_{\widetilde{\boldsymbol{z}},\widetilde{\boldsymbol{z}}})^{-1}\sigma_\varepsilon^{-2}\boldsymbol{P}(C^{\text{NS}}_{\widetilde{\boldsymbol{z}},\widetilde{\boldsymbol{z}}})^{-1} + C^{\text{NS}}_{\widetilde{\boldsymbol{z}},\widetilde{\boldsymbol{z}}}\right]\widetilde{\boldsymbol{z}}\right)\, d\widetilde{\boldsymbol{z}}.
\end{aligned} \tag{D.2}$$

We now notice that the terms inside the integral resemble the kernel of Gaussian with mean and variance

$$\begin{aligned}
\widehat{\boldsymbol{\mu}}_{\widetilde{\boldsymbol{z}}} &= \sigma_\varepsilon^{-2}C^{\text{NS}}_{\widetilde{\boldsymbol{z}},\widetilde{\boldsymbol{z}}}\left(\sigma_\varepsilon^{-2}\boldsymbol{P} + C^{\text{NS}}_{\widetilde{\boldsymbol{z}},\widetilde{\boldsymbol{z}}}\right)^{-1} B^T\boldsymbol{y}, \\
\widehat{\Sigma}_{\widetilde{\boldsymbol{z}}} &= \left((C^{\text{NS}}_{\widetilde{\boldsymbol{z}},\widetilde{\boldsymbol{z}}})^{-1}\sigma_\varepsilon^{-2}\boldsymbol{P}(C^{\text{NS}}_{\widetilde{\boldsymbol{z}},\widetilde{\boldsymbol{z}}})^{-1} + C^{\text{NS}}_{\widetilde{\boldsymbol{z}},\widetilde{\boldsymbol{z}}}\right)^{-1} \\
&= C^{\text{NS}}_{\widetilde{\boldsymbol{z}},\widetilde{\boldsymbol{z}}}\left(\sigma_\varepsilon^{-2}\boldsymbol{P} + C^{\text{NS}}_{\widetilde{\boldsymbol{z}},\widetilde{\boldsymbol{z}}}\right)^{-1} C^{\text{NS}}_{\widetilde{\boldsymbol{z}},\widetilde{\boldsymbol{z}}}.
\end{aligned}$$

Thus, by completing the square, we re-write Eq. (D.2) as

$$q(\widetilde{U}, \boldsymbol{\theta}, \boldsymbol{\varphi}) \propto \Xi |C^{\text{NS}}_{\widetilde{\boldsymbol{z}},\widetilde{\boldsymbol{z}}}|^{\frac{1}{2}} |\sigma_\varepsilon^{-2}\boldsymbol{P} + C^{\text{NS}}_{\widetilde{\boldsymbol{z}},\widetilde{\boldsymbol{z}}}|^{\frac{1}{2}} \exp\left(\frac{\sigma_\varepsilon^{-4}}{2}\left(\boldsymbol{y}^T B(\sigma_\varepsilon^{-2}\boldsymbol{P} + C^{\text{NS}}_{\widetilde{\boldsymbol{z}},\widetilde{\boldsymbol{z}}})^{-1} B^T \boldsymbol{y}\right)\right) \times \cancelto{1}{\int \text{N}(\widetilde{\boldsymbol{z}} | \widehat{\boldsymbol{\mu}}_{\widetilde{\boldsymbol{z}}}, \widehat{\Sigma}_{\widetilde{\boldsymbol{z}}})\, d\widetilde{\boldsymbol{z}}},}$$

where by plugging the values of Ξ and re-arranging terms, we obtain

$$q(\widetilde{U}, \boldsymbol{\theta}, \boldsymbol{\varphi}) \propto \sigma_\varepsilon^{-N} \pi(\boldsymbol{\theta}) \left(\prod_{d=1}^{D} \pi(\widetilde{\boldsymbol{u}}_{:d} \mid \boldsymbol{\varphi}_d)\pi(\boldsymbol{\varphi}_d)\right) \exp\left(\frac{1}{2\sigma_\varepsilon^4} \boldsymbol{y}^T B \left(C^{\text{NS}}_{\widetilde{\boldsymbol{z}},\widetilde{\boldsymbol{z}}} + \sigma_\varepsilon^{-2}\boldsymbol{P}\right)^{-1} B^T \boldsymbol{y}\right)$$
$$|C^{\text{NS}}_{\widetilde{\boldsymbol{z}},\widetilde{\boldsymbol{z}}} + \sigma_\varepsilon^{-2}\boldsymbol{P}|^{-\frac{1}{2}} |C^{\text{NS}}_{\widetilde{\boldsymbol{z}},\widetilde{\boldsymbol{z}}}|^{\frac{1}{2}} \exp\left(-\frac{1}{2\sigma_\varepsilon^2}\sum_{n=1}^{N}(y_n^2 + \tau_z^2) + \frac{1}{2\sigma_\varepsilon^2}\sum_{n=1}^{N}\alpha_n\right),$$

with

$$\begin{aligned}
\sum_{n=1}^{N} \alpha_n &= \sum_{n=1}^{N} \mathbb{E}_{\pi(\boldsymbol{u}_{n:}|\widetilde{U},\boldsymbol{\theta},\boldsymbol{\varphi})}[C^{\text{NS}}_{z_n,\widetilde{\boldsymbol{z}}}(C^{\text{NS}}_{\widetilde{\boldsymbol{z}},\widetilde{\boldsymbol{z}}})^{-1} C^{\text{NS}}_{\widetilde{\boldsymbol{z}},z_n}] \\
&= \sum_{i,j=1}^{M} (C^{\text{NS}}_{\widetilde{\boldsymbol{z}},\widetilde{\boldsymbol{z}}})^{-1}_{ij} \sum_{n=1}^{N} \mathbb{E}_{\pi(\boldsymbol{u}_{n:}|\widetilde{U},\boldsymbol{\theta},\boldsymbol{\varphi})}\left[C^{\text{NS}}_{\widetilde{z}_i,z_n} C^{\text{NS}}_{z_n,\widetilde{z}_j}\right] \\
&= \sum_{i,j=1}^{M} (C^{\text{NS}}_{\widetilde{\boldsymbol{z}},\widetilde{\boldsymbol{z}}})^{-1}_{ij} \boldsymbol{P}_{ij} \\
&= \sum_{i,j=1}^{M} \left((C^{\text{NS}}_{\widetilde{\boldsymbol{z}},\widetilde{\boldsymbol{z}}})^{-1} \odot \boldsymbol{P}\right)_{ij}.
\end{aligned}$$

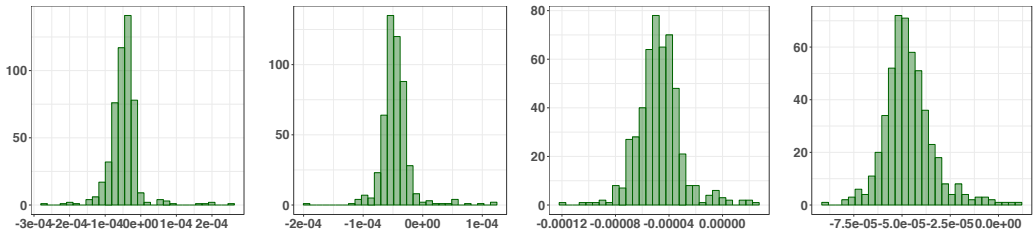
Moreover, the conditional variational posterior corresponds to $q(\widetilde{\boldsymbol{z}} \mid \boldsymbol{\theta}, \boldsymbol{\varphi}) = \text{N}(\widetilde{\boldsymbol{z}} \mid \widehat{\boldsymbol{\mu}}_{\widetilde{\boldsymbol{z}}}, \widehat{\Sigma}_{\widetilde{\boldsymbol{z}}})$, such that

$$q(\widetilde{\boldsymbol{z}} \mid \widetilde{U}, \boldsymbol{\theta}, \boldsymbol{\varphi}) = \text{N}\left(\sigma_\varepsilon^{-2} C^{\text{NS}}_{\widetilde{\boldsymbol{z}},\widetilde{\boldsymbol{z}}} \left(\sigma_\varepsilon^{-2}\boldsymbol{P} + C^{\text{NS}}_{\widetilde{\boldsymbol{z}},\widetilde{\boldsymbol{z}}}\right)^{-1} B^T \boldsymbol{y}, \sigma_\varepsilon^{-2} C^{\text{NS}}_{\widetilde{\boldsymbol{z}},\widetilde{\boldsymbol{z}}} \left(\sigma_\varepsilon^{-2}\boldsymbol{P} + C^{\text{NS}}_{\widetilde{\boldsymbol{z}},\widetilde{\boldsymbol{z}}}\right)^{-1} B^T \boldsymbol{y}\right)$$

The derivations of the isotropic and separable ARD cases are analogous to the computations above.

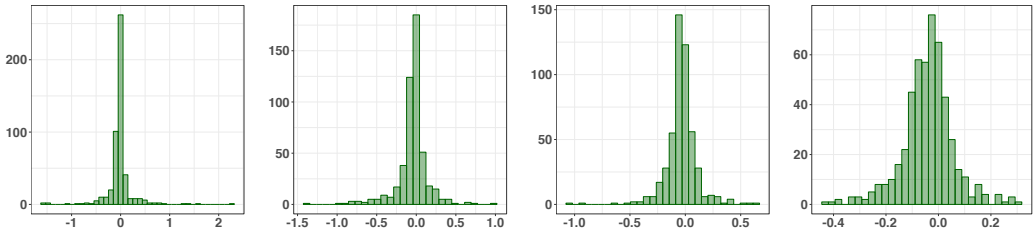
APPENDIX E

SUPPLEMENTARY MATERIAL FOR CHAPTER 6



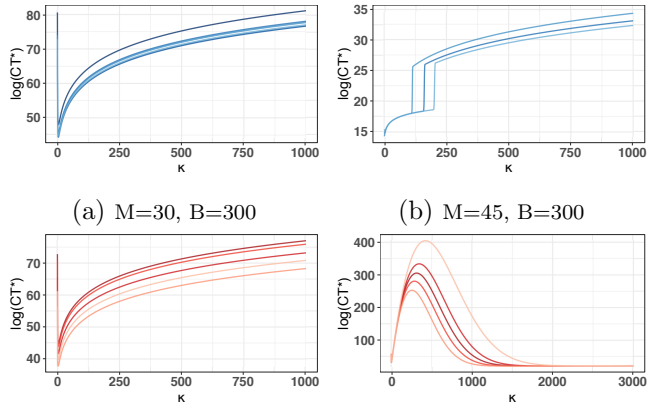
(a) M=30, B=30 (b) M=30, B=90 (c) M=30, B=150 (d) M=30, B=300

Figure E.1: Histograms of $\widehat{d}_B$ for $M = 30$ with different number of subsamples ($B = 30, 90, 150, 300$).



(a) M=45, B=30 (b) M=45, B=90 (c) M=45, B=150 (d) M=45, B=300

Figure E.2: Histograms of $\widehat{d}_B$ for $M = 45$ with different number of subsamples ($B = 30, 90, 150, 300$).



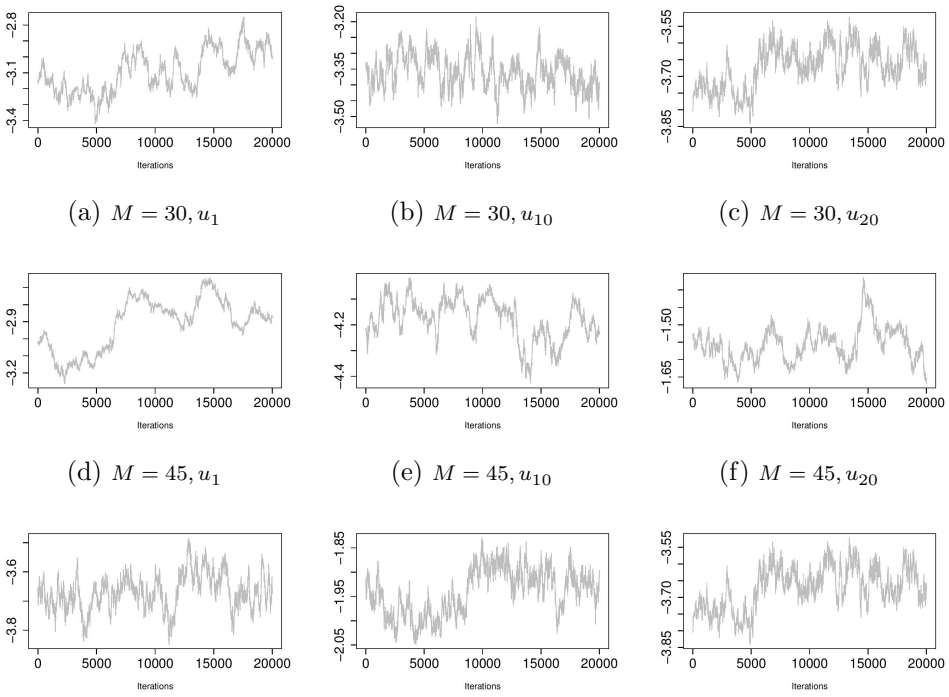
(a) M=30, B=300 (b) M=45, B=300

(c) M=30, B=300 (d) M=45, B=300

Figure E.3: Logarithm of the computational time measure $\mathrm{CT}^{\star}$ as a function of κ, for $M = 30$, $M = 45$ with $B = 300$. First column corresponds to two different small samples ($S = 100$) from the pilot MCMC for $M = 30$. Second column illustrates the results for $M = 45$. Each line showcase the results of a random subsample of size $B = 300$.

		Avg. time (min)
Full MCMC		15.13
M=30	J=4	1.17
	J=8	1.17
	J=10	0.83
	S-BP-PM	0.72
	S-BP-PM with MCWM	0.85
M=45	J=4	1.44
	J=8	1.60
	J=10	1.32
	S-BP-PM	1.61
	S-BP-PM with MCWM	1.94
M=60	J=4	4.09
	J=8	6.51
	J=10	8.18
	S-BP-PM	0.66
	S-BP-PM with MCWM	0.76

Table E.1: Computational time comparison. Average time (in minutes) required for 100 iterations of the MCMC. S-BP-PM shows the average time needed for $45,000$ iterations of Algorithm 14. S-BP-PM with MCWM presents the average over $50,000$ iterations, with the first $5,000$ iterations spent on the MCWM phase and $45,000$ in S-BP-PM.



(a) $M = 30, u_1$ (b) $M = 30, u_{10}$ (c) $M = 30, u_{20}$

(d) $M = 45, u_1$ (e) $M = 45, u_{10}$ (f) $M = 45, u_{20}$

(g) $M = 60, u_{10}$ (h) $M = 60, u_{30}$ (i) $M = 60, u_{60}$

Figure E.4: Traceplots after burnin period of some components of the logarithm of the spatially varying length-scale for $M = 30, 45, 60$.

GLOSSARY

AA ancillary augmentation.

AGP additive Gaussian process.

ARD automatic relevance determination.

block-m-Ell-SS block marginal elliptical slice sampler.

BNP Bayesian non-parametrics.

CDF cumulative distribution function.

CT computational time.

Ell-SS elliptical slice sampling.

EP expectation propagation.

ESS effective sample size.

GIMH grouped independence Metropolis–Hastings.

GMRF Gaussian Markov random field.

GP Gaussian process.

GPR Gaussian process regression.

HMC Hamiltonian Monte Carlo.

HPD highest posterior density.

INLA integrated nested Laplace approximation.

KL Kullback-Leibler.

LA Laplace approximation.

MAE mean absolute error.

MALA Metropolis-adjusted Langevin algorithm.

MCWM Monte Carlo within Metropolis.

m-Ell-SS marginal elliptical slice sampling.

MAP maximum a posteriori.

MCMC Markov chain Monte Carlo.

MH Metropolis-Hastings.

MWG Metropolis-within-Gibbs.

OES overall efficiency score.

pCN preconditioned Crank–Nicolson.

PM pseudo-marginal.

RW-MH random walk Metropolis-Hastings.

S-BP-PM signed block-Poisson pseudo-marginal.

SE squared exponential.

SPDE stochastic partial differential equation.

w-Ell-SS whitened elliptical slice sampling.

BIBLIOGRAPHY

Abrahamsen, P. (1997). A review of Gaussian random fields and correlation functions. Technical report, Norwegian Computing Center.

Álvarez, M. A., Rosasco, L., Lawrence, N. D., et al. (2012). Kernels for vector-valued functions: A review. *Foundations and Trends® in Machine Learning 4*(3), 195–266.

Anderes, E. B. and Stein, M. L. (2008). Estimating deformations of isotropic Gaussian random fields on the plane. *The Annals of Statistics 36*(2), 719–741.

Andrieu, C., Roberts, G. O., et al. (2009). The pseudo-marginal approach for efficient Monte Carlo computations. *The Annals of Statistics 37*(2), 697–725.

Beaumont, M. A. (2003). Estimation of population growth or decline in genetically monitored populations. *Genetics 164*(3), 1139–1160.

Berrocal, V. J., Raftery, A. E., Gneiting, T., and Steed, R. C. (2010). Probabilistic weather forecasting for winter road maintenance. *Journal of the American Statistical Association 105*(490), 522–537.

Beskos, A., Roberts, G., Stuart, A., and Voss, J. (2008). MCMC methods for diffusion bridges. *Stochastics and Dynamics 8*(03), 319–350.

Blei, D. M., Kucukelbir, A., and McAuliffe, J. D. (2017). Variational inference: a review for statisticians. *Journal of the American Statistical Association 112*(518), 859–877.

Blocker, A. W. (2018). *fastGHQuad: Fast 'Rcpp' Implementation of Gauss-Hermite Quadrature*. R package version 1.0.

Blomqvist, K., Kaski, S., and Heinonen, M. (2018). Deep convolutional Gaussian processes. arXiv preprint arXiv:1810.03052.

Brooks, S., Gelman, A., Jones, G., and Meng, X.-L. (2011). *Handbook of Markov chain Monte Carlo*. CRC press.

Buche, D., Schraudolph, N. N., and Koumoutsakos, P. (2005). Accelerating evolutionary algorithms with Gaussian process fitness function models. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews) 35*(2), 183–194.

Buja, A., Hastie, T., and Tibshirani, R. (1989). Linear smoothers and additive models. *The Annals of Statistics 17*(2), 453–510.

Carnell, R. (2016). *lhs: Latin Hypercube Samples*. R package version 0.13.

Chen, V., Dunlop, M. M., Papaspiliopoulos, O., and Stuart, A. M. (2019). Dimension-robust MCMC in Bayesian inverse problems. arXiv preprint arXiv:1803.03344v2.

Cheng, L., Ramchandran, S., Vatanen, T., Lietzén, N., Lahesmaa, R., Vehtari, A., and Lähdesmäki, H. (2019). An additive gaussian process regression model for interpretable non-parametric analysis of longitudinal data. *Nature Communications 10*(1), 1798.

Christensen, O. F., Roberts, G. O., and Sköld, M. (2006). Robust markov chain monte carlo methods for spatial generalized linear mixed models. *Journal of Computational and Graphical Statistics 15*(1), 1–17.

Cotter, S. L., Roberts, G. O., Stuart, A. M., and White, D. (2013). MCMC methods for functions: modifying old algorithms to make them faster. *Statistical Science 28*(3), 424–446.

Cutajar, K., Pullin, M., Damianou, A., Lawrence, N., and González, J. (2019). Deep Gaussian processes for multi-fidelity modeling. arXiv preprint arXiv:1903.07320.

Damianou, A. and Lawrence, N. (2013). Deep Gaussian processes. In *Artificial Intelligence and Statistics*, 207–215.

Damien, P., Dellaportas, P., Polson, N. G., and Stephens, D. A. (2013). *Bayesian theory and applications*. OUP Oxford.

Datta, A., Banerjee, S., Finley, A. O., and Gelfand, A. E. (2016). Hierarchical nearest-neighbor Gaussian prfocess models for large geostatistical datasets. *Journal of the American Statistical Association 111*(514), 800–812.

Donoho, D. L. and Johnstone, I. M. (1995). Adapting to unknown smoothness via wavelet shrinkage. *Journal of the American Statistical Association 90*(432), 1200–1224.

Drovandi, C. C., Moores, M. T., and Boys, R. J. (2018). Accelerating pseudo-marginal mcmc using gaussian processes. *Computational Statistics & Data Analysis 118*, 1–17.

Duane, S., Kennedy, A. D., Pendleton, B. J., and Roweth, D. (1987). Hybrid Monte Carlo. *Physics letters B 195*(2), 216–222.

Dunlop, M. M., Girolami, M., Stuart, A. M., and Teckentrup, A. L. (2018). How deep are deep Gaussian processes? *Journal of Machine Learning Research 19*, 1–46.

Durrande, N., Adam, V., Bordeaux, L., Eleftheriadis, S., and Hensman, J. (2019). Banded matrix operators for Gaussian Markov models in the automatic differentiation era. In *Artificial Intelligence and Statistics*.

Duvenaud, D., Rippel, O., Adams, R., and Ghahramani, Z. (2014). Avoiding pathologies in very deep networks. In *Artificial Intelligence and Statistics*, 202–210.

Duvenaud, D. K., Nickisch, H., and Rasmussen, C. E. (2011). Additive Gaussian processes. In *Advances in Neural Information Processing Systems*, 226–234.

Fagan, F., Bhandari, J., and Cunningham, J. (2016). Elliptical slice sampling with expectation propagation. In *Uncertainty in Artificial Intelligence*.

Fearnhead, P., Papaspiliopoulos, O., Roberts, G. O., and Stuart, A. (2010). Random-weight particle filtering of continuous time processes. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 72*(4), 497–512.

Filippone, M. and Girolami, M. (2014). Pseudo-marginal Bayesian inference for Gaussian processes. *IEEE Transactions on Pattern Analysis and Machine Intelligence 36*(11), 2214–2226.

Filippone, M., Zhong, M., and Girolami, M. (2013). A comparative evaluation of stochastic-based inference methods for Gaussian process models. *Machine Learning 93*(1), 93–114.

Finkenstadt, B., Held, L., and Isham, V. (2006). *Statistical methods for spatio-temporal systems*. Chapman and Hall–CRC.

Finley, A. O., Datta, A., Cook, B. D., Morton, D. C., Andersen, H. E., and Banerjee, S. (2019). Efficient algorithms for bayesian nearest neighbor gaussian processes. *Journal of Computational and Graphical Statistics 28*(2), 401–414.

Fouedjio, F. (2016). Second-order non-stationary modeling approaches for univariate geostatistical data. *Stochastic Environmental Research and Risk Assessment 31*(8), 1887–1906.

Fouedjio, F., Desassis, N., and Rivoirard, J. (2016). A generalized convolution model and estimation for non-stationary random functions. *Spatial Statistics 16*, 35–52.

Fox, E. and Dunson, D. B. (2012). Multiresolution Gaussian processes. In *Advances in Neural Information Processing Systems*, 737–745.

Friedman, J. H. and Stuetzle, W. (1981). Projection pursuit regression. *Journal of the American Statistical Association 76*(376), 817–823.

Fuglstad, G.-A., Lindgren, F., Simpson, D., and Rue, H. (2015). Exploring a new class of non-stationary spatial Gaussian random fields with varying local anisotropy. *Statistica Sinica 25*(1), 115–133.

Fuglstad, G.-A., Simpson, D., Lindgren, F., and Rue, H. (2015). Does non-stationary spatial data always require non-stationary random fields. *Spatial Statistics 14*, 505–531.

Fuglstad, G.-A., Simpson, D., Lindgren, F., and Rue, H. (2019). Constructing priors that penalize the complexity of gaussian random fields. *Journal of the American Statistical Association 114*(525), 445–452.

Gadd, C., Wade, S., Shah, A., and Grammatopoulos, D. (2018). Pseudo-marginal bayesian inference for supervised gaussian process latent variable models. arXiv preprint arXiv:1803.10746.

Gelfand, A. E., Diggle, P., Guttorp, P., and Fuentes, M. (2010). *Handbook of Spatial Statistics*. CRC press.

Gelman, A., Carlin, J. B., Stern, H. S., and Rubin, D. B. (2014). *Bayesian Data Analysis*, Volume 2. Taylor & Francis.

Geman, S. and Geman, D. (1987). Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. In *Readings in Computer Vision*, 564–584. Elsevier.

Genton, M. G. (2007). Separable approximations of space-time covariance matrices. *Environmetrics 18*(7), 681–695.

Gibbs, M. N. (1997). *Bayesian Gaussian processes for regression and classification.* Ph. D. thesis, Department of Physics, University of Cambridge.

Gilboa, E., Saatçi, Y., and Cunningham, J. (2013). Scaling multidimensional gaussian processes using projected additive approximations. In *International Conference on Machine Learning*, 454–461.

Gilboa, E., Saatçi, Y., and Cunningham, J. P. (2015). Scaling multidimensional inference for structured Gaussian processes. *IEEE Transactions on Pattern Analysis and Machine Intelligence 37*(2), 424–436.

Gilks, W. R., Richardson, S., and Spiegelhalter, D. (1995). *Markov chain Monte Carlo in practice.* Chapman and Hall/CRC.

Girolami, M. and Calderhead, B. (2011). Riemann manifold Langevin and Hamiltonian Monte Carlo methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 73*(2), 123–214.

Golub, G. H. and Van Loan, C. F. (1996). *Matrix Computations* (Third ed.). The Johns Hopkins University Press.

Gramacy, R. B. (2007). tgp: an R package for Bayesian nonstationary, semiparametric nonlinear regression and design by treed Gaussian process models. *Journal of Statistical Software 19*(9), 1–46.

Gramacy, R. B. and Lee, H. K. (2012). Bayesian treed Gaussian process models with an application to computer modeling. *Journal of the American Statistical Association 103*, 1119–1130.

Harrison, J. (2009). Fast and accurate bessel function computation. In *2009 19th IEEE Symposium on Computer Arithmetic*, 104–113. IEEE.

Harville, D. A. (1997). *Matrix algebra from a statistician's perspective*, Volume 1. Springer.

Hastings, W. K. (1970). Monte Carlo sampling methods using Markov chains and their applications. *Biometrika 57*(1), 97–109.

Heaton, t. J., Datta, A., Finley, A. O., Furrer, R., Guinness, J., Guhaniyogi, R., Gerber, F., Gramacy, R. B., Hammerling, D., Katzfuss, M., Lindgren, F., Nychka, D. W., Sun, F., and Zammit-Mangion, A. (2018). A case study competition among methods for analyzing large spatial data. *Journal of Agricultural, Biological and Environmental Statistics*.

Hegde, P., Heinonen, M., Lähdesmäki, H., and Kaski, S. (2019). Deep learning with differential Gaussian process flows. In *Artificial Intelligence and Statistics*, Volume 89, 1812–1821.

Heinonen, M., Mannerström, H., Rousu, J., Kaski, S., and Lähdesmäki, H. (2016). Non-stationary Gaussian Process Regression with Hamiltonian Monte Carlo. In *Artificial Intelligence and Statistics*, Volume 51, 732–740.

Hensman, J., Fusi, N., and Lawrence, N. D. (2013). Gaussian processes for big data. In *Proceedings of the Twenty-Ninth Conference on Uncertainty in Artificial Intelligence*, UAI'13, 282–290. AUAI Press.

Hensman, J., Matthews, A., Filippone, M., and Ghahramani, Z. (2015). MCMC for variationally sparse Gaussian processes. In *Advances in Neural Information Processing Systems*, 1648–1656.

Higdon, D. (1998). A process-convolution approach to modelling temperatures in the North Atlantic Ocean. *Environmental and Ecological Statistics 5*(2), 173–190.

Higdon, D., Swall, J., and Kern, J. (1999). Non-stationary spatial modeling. *Bayesian statistics 6*(1), 761–768.

Hoffman, M. D. and Gelman, A. (2014). The no-u-turn sampler: adaptively setting path lengths in hamiltonian monte carlo. *Journal of Machine Learning Research 15*(1), 1593–1623.

Holland, D. M., Saltzman, N., Cox, L. H., and Nychka, D. (1999). Spatial prediction of sulfur dioxide in the Eastern United States. In *geoENV II—Geostatistics for Environmental Applications*, Volume 10, 65–76. Springer.

Kaipio, J. and Somersalo, E. (2006). *Statistical and computational inverse problems*. Springer Science & Business Media.

Karny, M. (2006). *Optimized Bayesian dynamic advising: Theory and algorithms*. Springer Science & Business Media.

Katzfuss, M. (2013). Bayesian nonstationary spatial modeling for very large datasets. *Environmetrics 24*(3), 189–200.

Kim, H.-M., Mallick, B. K., and Holmes, C. (2005). Analyzing nonstationary spatial data using piecewise Gaussian processes. *Journal of the American Statistical Association 100*(470), 653–668.

Kimeldorf, G. S. and Wahba, G. (1970). A correspondence between bayesian estimation on stochastic processes and smoothing by splines. *The Annals of Mathematical Statistics 41*(2), 495–502.

Kleiber, W. and Nychka, D. (2012). Nonstationary modeling for multivariate spatial processes. *Journal of Multivariate Analysis 112*, 76–91.

Kuss, M. and Rasmussen, C. E. (2006). Assessing approximations for Gaussian process classification. In *Advances in Neural Information Processing Systems*, 699–706.

Lang, T., Plagemann, C., and Burgard, W. (2007). Adaptive Non-Stationary Kernel Regression for Terrain Modeling. In *Robotics: Science and Systems*.

Lawrence, N., Seeger, M., and Herbrich, R. (2003). Fast sparse Gaussian process ethods: the informative vector machine. In *Proceedings of the 16th Annual Conference on Neural Information Processing Systems*, 609–616.

Lawrence, N. D. (2004). Gaussian process latent variable models for visualisation of high dimensional data. In *Advances in Neural Information Processing Systems*, 329–336.

Liang, F., Liu, C., and Carroll, R. (2011). *Advanced Markov chain Monte Carlo methods: learning from past samples*, Volume 714. John Wiley & Sons.

Lindgren, F., Rue, H., and Lindström, J. (2011). An explicit link between Gaussian fields and Gaussian Markov random fields: the stochastic partial differential equation approach. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 73*(4), 423–498.

Liu, H., Ong, Y.-S., Shen, X., and Cai, J. (2018). When Gaussian process meets big data: A review of scalable GPs. *arXiv preprint arXiv:1807.01065*.

Lyne, A.-M., Girolami, M., Atchadé, Y., Strathmann, H., Simpson, D., et al. (2015). On Russian roulette estimates for Bayesian inference with doubly-intractable likelihoods. *Statistical science 30*(4), 443–467.

MacKay, D. J. (1996). Bayesian methods for backpropagation networks. In *Models of neural networks III*, 211–254. Springer.

Matheron, G. (1973). The intrinsic random functions and their applications. *Advances in Applied Probability*, 439–468.

Matthews, A. G., van der Wilk, M., Nickson, T., Fujii, K., Boukouvalas, A., Leoón-Villagrá, P., Ghahramani, Z., and Hensman, J. (2017). GPflow: A Gaussian process library using TensorFlow. *Journal of Machine Learning Research 18*(1), 1299–1304.

Matthews, A. G. d. G., Hensman, J., Turner, R., and Ghahramani, Z. (2016). On sparse variational methods and the Kullback-Leibler divergence between stochastic processes. *Journal of Machine Learning Research 51*, 231–239.

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., and Teller, E. (1953). Equation of state calculations by fast computing machines. *The Journal of Chemical Physics 21*(6), 1087–1092.

Minka, T. P. (2001). Expectation propagation for approximate Bayesian inference. In *Proceedings of the Seventeenth Conference on Uncertainty in Artificial Intelligencen*, 362–369. Morgan Kaufmann Publishers Inc.

Montagna, S. and Tokdar, S. T. (2016). Computer emulation with nonstationary Gaussian processes. *SIAM/ASA Journal on Uncertainty Quantification 4*(1), 26–47.

Monterrubio-Gómez, K., Roininen, L., Wade, S., Damoulas, T., and Girolami, M. (2019). Posterior Inference for Sparse Hierarchical Non-stationary Models. arXiv:1804.01431.

Moore, D. and Russell, S. J. (2015). Gaussian process random fields. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett (Eds.), *Advances in Neural Information Processing Systems 28*, 3357–3365. Curran Associates, Inc.

Murray, I., Adams, R., and MacKay, D. (2010). Elliptical slice sampling. In *Artificial Intelligence and Statistics*, 541–548.

Murray, I. and Adams, R. P. (2010). Slice sampling covariance hyperparameters of latent Gaussian models. In *Advances in Neural Information Processing Systems*, 1732–1740.

Murray, I. and Graham, M. (2016). Pseudo-marginal slice sampling. In *Artificial Intelligence and Statistics*, 911–919.

Neal, R. M. (1995). *Bayesian learning for neural networks*. Ph. D. thesis, Department of Computer Science, University of Toronto.

Neal, R. M. (1997). Monte carlo implementation of Gaussian process models for bayesian regression and classification. *arXiv preprint physics/9701026*.

Neal, R. M. et al. (2003). Slice sampling. *The Annals of Statistics 31*(3), 705–767.

Neto, J. H. V., Schmidt, A. M., and Guttorp, P. (2014). Accounting for spatially varying directional effects in spatial covariance structures. *Journal of the Royal Statistical Society: Series C (Applied Statistics) 63*(1), 103–122.

Nychka, D., Wikle, C., and Royle, J. A. (2002). Multiresolution models for nonstationary spatial covariance functions. *Statistical Modelling 2*(4), 315–331.

O'Hagan, A. (1991). Bayes–Hermite quadrature. *Journal of Statistical Planning and Inference 29*(3), 245–260.

Opper, M. and Archambeau, C. (2009). The variational Gaussian approximation revisited. *Neural computation 21*(3), 786–792.

Paciorek, C. J. (2003). *Nonstationary Gaussian processes for regression and spatial modelling*. Ph. D. thesis, Department of Statistics, Carnegie Mellon University.

Paciorek, C. J. and Schervish, M. J. (2006). Spatial modelling using a new class of nonstationary covariance functions. *Environmetrics 17*(5), 483–506.

Papaspiliopoulos, O., Roberts, G. O., and Sköld, M. (2007). A general framework for the parametrization of hierarchical models. *Statistical Science 22*(1), 59–73.

Pintore, A. and Holmes, C. (2004). Spatially adaptive non-stationary covariance functions via spatially adaptive spectra. Technical report, University of Oxford.

Press, W. H., Flannery, B. P., Teukolsky, S. A., and Vetterling, W. T. (1988). *Numerical Recipes in C: The Art of Scientific Computing*. New York, NY, USA: Cambridge University Press.

Quiroz, M., Tran, M.-N., Villani, M., Kohn, R., and Dang, K.-D. (2018). The block-Poisson estimator for optimally tuned exact subsampling MCMC. arXiv:1603.08232v5.

Raftery, A. E. and Lewis, S. M. (1992). [Practical Markov chain Monte Carlo]: Comment: One long run with diagnostics: Implementation strategies for Markov chain Monte Carlo. *Statistical science 7*(4), 493–497.

Rasmussen, C. and Williams, C. (2006). *Gaussian processes for machine learning*. Adaptative Computation and Machine Learning Series. the MIT Press.

Rattray, M., Yang, J., Ahmed, S., and Boukouvalas, A. (2019). *Modelling gene expression dynamics with Gaussian process inference*, Chapter 31, 879–20. John Wiley & Sons, Ltd.

Risser, M. D. and Calder, C. A. (2015). Regression-based covariance functions for nonstationary spatial modeling. *Environmetrics 26*(4), 284–297.

Risser, M. D. and Calder, C. A. (2017). Local likelihood estimation for covariance functions with spatially-varying parameters: The convoSPAT package for R. *Journal of Statistical Software 81*(1), 1–32.

Robert, C. and Casella, G. (2009). *Introducing Monte Carlo methods with R*. Springer Science & Business Media.

Robert, C. and Casella, G. (2013). *Monte Carlo Statistical Methods*. Springer Science & Business Media.

Roberts, G. O., Gelman, A., Gilks, W. R., et al. (1997). Weak convergence and optimal scaling of random walk Metropolis algorithms. *The Annals of Applied Probability 7*(1), 110–120.

Roberts, G. O. and Rosenthal, J. S. (1998). Optimal scaling of discrete approximations to Langevin diffusions. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 60*(1), 255–268.

Roberts, G. O. and Rosenthal, J. S. (2009). Examples of adaptive MCMC. *Journal of Computational and Graphical Statistics 18*(2), 349–367.

Roberts, G. O., Tweedie, R. L., et al. (1996). Exponential convergence of Langevin distributions and their discrete approximations. *Bernoulli 2*(4), 341–363.

Roininen, L., Girolami, M., Lasanen, S., and Markkanen, M. (2019). Hyperpriors for Matérn fields with applications in Bayesian inversion. *Inverse Problems & Imaging 13*, 1.

Rougier, J. (2017). A representation theorem for stochastic processes with separable covariance functions, and its implications for emulation. arXiv preprint arXiv:1702.05599.

Rozanov, J. A. (1977). Markov random fields and stochastic partial differential equations. *Mathematics of the USSR-Sbornik 32*(4), 515.

Rue, H. and Held, L. (2005). *Gaussian Markov random fields: Theory and applications*. Chapman and Hall/CRC.

Rue, H., Martino, S., and Chopin, N. (2009). Approximate Bayesian inference for latent Gaussian models by using integrated nested Laplace approximations. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 71*(2), 319–392.

Rutishauser, H. (1962). On a modification of the QD-algorithm with Graeffe-type convergence. *Zeitschrift für angewandte Mathematik und Physik ZAMP 13*(5), 493–496.

Sampson, P., Damian, D., and Guttorp, P. (2001). Advances in modeling and inference for environmental processes with nonstationary spatial covariance. In *GeoENV III—Geostatistics for Environmental Applications*, Volume 11 of *Quantitative Geology and Geostatistics*, 17–32. Springer.

Schmidt, A. M. and O'Hagan, A. (2003). Bayesian inference for non-stationary spatial covariance structure via spatial deformations. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 65*(3), 743–758.

Seeger, M. (2000). Relationships between Gaussian processes, support vector machines and smoothing splines. Technical report, University of Edinburgh.

Seeger, M. (2004). Gaussian processes for machine learning. *International Journal of Neural Systems 14*(02), 69–106.

Seiler, M. C. and Seiler, F. A. (1989). Numerical recipes in C: the art of scientific computing. *Risk Analysis 9*(3), 415–416.

Sherman, M. (2011). *Spatial statistics and spatio-temporal data: covariance functions and directional properties*. John Wiley & Sons.

Snelson, E. and Ghahramani, Z. (2006). Sparse Gaussian processes using pseudo-inputs. In *Advances in Neural Information Processing Systems*, 1257–1264.

Sollich, P. (2002). Bayesian methods for support vector machines: Evidence and predictive class probabilities. *Machine learning 46*(1-3), 21–52.

Stathopoulos, V., Zamora-Gutierrez, V., Jones, K., and Girolami, M. (2014). Bat call identification with Gaussian process multinomial probit regression and a dynamic time warping kernel. In *Artificial Intelligence and Statistics*, 913–921.

Stein, M. L. (2005). Nonstationary spatial covariance functions. Technical report, Center for Integrating Statistical and Environmental Science, University of Chicago, Chicago.

Stein, M. L. (2012). *Interpolation of spatial data: Some theory for Kriging*. Springer Science & Business Media.

Stephenson, J., Holmes, C., Gallagher, K., and Pintore, A. (2005). A statistical technique for modelling non-stationary spatial processes. In *Geostatistics Banff 2004*, 125–134. Springer.

Stroud, A. H. and Secrest, D. (1966). *Gaussian quadrature formulas*. Prentice-Hall.

Titsias, M. (2009). Variational learning of inducing variables in sparse gaussian processes. In *Artificial Intelligence and Statistics*, 567–574.

Titsias, M. and Lázaro-Gredilla, M. (2013). Variational inference for mahalanobis distance metrics in gaussian process regression. In *Advances in Neural Information Processing Systems*, 279–287.

Titsias, M. K. and Papaspiliopoulos, O. (2018). Auxiliary gradient-based sampling algorithms. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 80*, 749–767.

Titsias, M. K., Rattray, M., and Lawrence, N. D. (2011). *Markov chain Monte Carlo algorithms for Gaussian processes*, 295–316. Cambridge University Press.

Vanhatalo, J., Riihimäki, J., Hartikainen, J., Jylänki, P., Tolvanen, V., and Vehtari, A. (2015). Bayesian modeling with Gaussian processes using the GPstuff toolbox.

Vanhatalo, J. and Vehtari, A. (2007). Sparse log Gaussian processes via MCMC for spatial epidemiology. In *Gaussian Processes in Practice*, Volume 1, 73–89.

Vannucci, M. and Corradi, F. (1999). Covariance structure of wavelet coefficients: Theory and models in a Bayesian perspective. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 61*(4), 971–986.

Volodina, V. and Williamson, D. B. (2018). Diagnostic-driven nonstationary emulators using kernel mixtures. arXiv preprint arXiv:1803.04906.

Williams, C. and Seeger, M. (2001). Using the Nyström method to speed up kernel machines. In *Proceedings of the 14th Annual Conference on Neural Information Processing Systems*, 682–688. MIT Press.

Williams, C. K. (1998). Computation with infinite neural networks. *Neural Computation 10*(5), 1203–1216.

Williams, C. K. and Barber, D. (1998). Bayesian classification with Gaussian processes. *IEEE Transactions on Pattern Analysis and Machine Intelligence 20*(12), 1342–1351.

Xiong, X., Šmídl, V., and Filippone, M. (2017). Adaptive multiple importance sampling for Gaussian processes. *Journal of Statistical Computation and Simulation 87*(8), 1644–1665.

Xiong, Y., Chen, W., Apley, D., and Ding, X. (2007). A non-stationary covariance-based Kriging method for metamodelling in engineering design. *International Journal for Numerical Methods in Engineering 71*(6), 733–756.

Yang, J., Penfold, C. A., Grant, M. R., and Rattray, M. (2016). Inferring the perturbation time from biological time course data. *Bioinformatics 32*(19), 2956–2964.

Yu, Y. and Meng, X.-L. (2011). To center or not to center: That is not the question -An ancillarity-sufficiency interweaving strategy (ASIS) for boosting MCMC efficiency. *Journal of Computational and Graphical Statistics 20*(3), 531–570.

Yue, Y. R., Simpson, D., Lindgren, F., and Rue, H. (2014). Bayesian adaptive smoothing splines using stochastic differential equations. *Bayesian Analysis 9*(2), 397–424.

Zhang, H. (2004). Inconsistent estimation and asymptotically equal interpolations in model-based geostatistics. *Journal of the American Statistical Association 99*(465), 250–261.